

# Index

## HTTP Status Codes

- 100 Continue code, 400–402, 410–411
- 101 Switching Protocols code, 411
- 110 Response is stale code, 503
- 111 Revalidation failed code, 503
- 112 Disconnected operation code, 503
- 113 Heuristic expiration code, 503
- 199 Miscellaneous warning code, 503
- 200 OK code, 411
- 201 Created code, 411–412
- 202 Accepted code, 412
- 203 Non-authoritative Information code, 412
- 204 No Content code, 412–413
- 205 Reset Content code, 413
- 206 Partial Content code, 413–414
- 214 Transformation applied code, 503–504
- 299 Miscellaneous persistent warning code, 504
- 300 Multiple Choices code, 414–415
- 301 Moved Permanently code, 415
- 302 Found code, 415–416
- 303 See Other code, 416
- 304 Not Modified code, 416–417
- 305 Use Proxy code, 417
- 307 Temporary Redirect code, 417–418
- 400 Bad Request code, 418
- 401 Unauthorized code, 186, 418–419
- 403 Forbidden code, 186, 419
- 404 Not Found code, 419
- 405 Method Not Allowed code, 419
- 406 Not Acceptable code, 419–420
- 407 Proxy Authentication Required code, 186, 420
- 408 Request Timeout code, 420
- 409 Conflict code, 420–421
- 410 Gone code, 421
- 411 Length Required code, 421
- 412 Precondition Failed code, 421
- 413 Request Entity Too Large code, 422
- 414 Request-URI Too Long code, 422
- 415 Unsupported Media Type code, 422
- 416 Requested Range Not Satisfiable code, 422
- 417 Expectation Failed code, 423
- 500 Internal Server Error code, 423
- 501 Not Implemented code, 423
- 502 Bad Gateway code, 423
- 503 Service Unavailable code, 423–424
- 504 Gateway Timeout code, 424
- 505 HTTP Version Not Supported code, 424

## # (signs)

- for comments, 25
- HTTP rules, 367

## A

### AAA (Access, Authentication, and Authorization), 177

- access denial in, 186–187
- in Apache 1.x/2.0, 182
- in Apache 2.1/2.2, 182–185
- authentication in, 186–187
- logic in, 185–186
- login schemes in, 195–199

- AAA (Access, Authentication, and Authorization), *continued***
  - modules for, 187–188
    - authentication function*, 190–192
    - basic authentication providers*, 188–190, 193–195
    - configuration*, 193
    - digest authentication providers*, 193–195
  - overview, 180–181
  - summary, 199–200
- Abnormal running, modules dealing with, 337–338**
- absoluteURI option, 389**
- abstract in HTTP/1.1 specification, 358**
- Accept request-header field, 159, 453–455, 507**
- Accept-Charset request-header field, 455–456**
- Accept-Encoding request-header field, 456–457**
- Accept-Language request-header field, 159, 382, 457–458**
- Accept-Ranges response-header field, 459**
- Accepted status code, 412**
- Access**
  - AAA. *See* AAA (Access, Authentication, and Authorization)
  - denying, 186–187
  - in HTTP/1.1 specification, 424
- access\_checker hook**
  - in host-based access control, 183
  - purpose, 45
  - in request processing, 155–156
- ACCESS\_CONF option, 246**
- Acknowledgments in HTTP/1.1 specification, 510–512, 530**
- AddHandler directive, 42**
- AddLanguage directive, 159**
- AddOutputFilter directive, 211**
- AddOutputFilterByType directive, 211**
- Age calculations in cache expiration model, 434–436**
- Age of responses in HTTP/1.1 specification, 363**
- Age response-header field, 459**
- Agent-driven content negotiation, 425–427**
- Alias directive, 157–158**
- Allow directive, 182**
- Allow entity-header field, 459–460**
- Ampersands (&) for key/value pairs, 138**
- Anonymous authentication, 193**
- ap\_header files, 39**
- ap\_check\_cmd\_context function, 255**
- ap\_check\_command\_context function, 256**
- ap\_config.h file, 268**
- ap\_dbd API, 290, 302–303**
- ap\_dbd\_acquire function, 283–284, 286–287, 302–303**
- ap\_dbd\_cacquire function, 287, 302–303**
- ap\_dbd\_close function, 283, 287, 302**
- ap\_dbd\_open function, 282, 287, 302–303**
- ap\_dbd\_prepare function, 287, 302, 304**
- ap\_dbd\_t type, 302**
- ap\_destroy\_sub\_req function, 164, 167**
- AP\_FILTER\_PROTO\_CHANGE field, 216**
- AP\_FILTER\_PROTO\_CHANGE\_LENGTH field, 216**
- AP\_FILTER\_PROTO\_NO\_BYTERANGE field, 216**
- AP\_FILTER\_PROTO\_NO\_CACHE field, 217**
- AP\_FILTER\_PROTO\_NO\_PROXY field, 217**
- AP\_FILTER\_PROTO\_TRANSFORM field, 217**
- ap\_filter\_protocol function, 216**
- ap\_filter\_rec\_t type, 209–210**
- ap\_filter\_t type, 67, 208–209**
- AP\_FTYPE\_CONNECTION filters, 204**
- AP\_FTYPE\_CONTENT\_SET filters, 203–204**
- AP\_FTYPE\_NETWORK filters, 204**
- AP\_FTYPE\_PROTOCOL filters, 203**
- AP\_FTYPE\_RESOURCE filters, 203–204**
- AP\_FTYPE\_TRANSCODE filters, 204**
- ap\_get\_brigade function, 137**

- ap\_get\_module\_config function, 144
- ap\_getword\_conf function, 250
- ap\_hook\_access\_checker hook, 181
- ap\_hook\_auth\_checker hook, 181
- ap\_hook\_check\_user\_id hook, 181
- ap\_hook\_fatal\_exception hooks, 336
- ap\_hook\_handler function, 268–269
- ap\_hook\_monitor function, 337–338
- ap\_hook\_translate\_name function, 268
- ap\_hook\_type\_checker function, 269
- AP\_IMPLEMENT\_EXTERNAL\_HOOK\_RU
  - N\_ALL macro, 268
- AP\_IMPLEMENT\_EXTERNAL\_HOOK\_RU
  - N\_FIRST macro, 268
- AP\_IMPLEMENT\_EXTERNAL\_HOOK\_VOI
  - D macro, 268
- AP\_INIT\_FLAG macro, 246–247
- AP\_INIT\_ITERATE macro, 247–248
- AP\_INIT\_ITERATE2 macro, 247–250
- AP\_INIT\_NO\_ARGS macro, 246
- AP\_INIT\_RAW\_ARGS macro, 247, 250, 258
- AP\_INIT\_TAKE1 macro, 242–243, 246
- AP\_INIT\_TAKE2 macro, 247
- AP\_INIT\_TAKE3 macro, 247
- AP\_INIT\_TAKE12 macro, 247
- ap\_internal\_fast\_redirect function, 162
- ap\_internal\_redirect function, 161
- ap\_log\_cerror function, 324–325
- ap\_log\_error function, 324–325
- ap\_log\_perror function, 324–325
- ap\_log\_rerror function, 324–325
- ap\_lookup\_provider function, 275
- ap\_meets\_conditions function, 146
- AP\_MODE\_GETLINE mode, 231
- AP\_MODE\_READBYTES mode, 231
- ap\_pass\_brigade function, 137, 205, 221
- ap\_pcfg\_open\_custom function, 261
- ap\_provider API, 272–277
- ap\_provider.h header file, 41
- ap\_register\_output\_filter\_protocol function, 216
- ap\_run\_sub\_req function, 164
- ap\_set\_content\_type function, 125, 211
- ap\_set\_file\_slot function, 245
- ap\_set\_flag\_slot function, 247
- ap\_set\_module\_config function, 283
- ap\_some\_auth\_required function, 186
- ap\_sub\_req\_lookup\_dirent function, 164
- ap\_sub\_req\_lookup\_file function, 164
- ap\_sub\_req\_lookup\_uri function, 164
- ap\_sub\_req\_method\_uri function, 164
- Apache 1
  - AAA in, 182
  - history, 1–2
- Apache 2
  - AAA in, 182
  - connection pooling in, 291–292
  - history, 2–3
- Apache 2.2
  - AAA in, 182–185
  - smart filtering in, 211–217
- Apache Bugzilla database, 10
- #apache chat channel, 17
- Apache license, 12–14, 343
  - accepting warranty or additional liability conditions, 347
  - applying, 347–348
  - definitions, 343–345
  - disclaimer of warranty, 346–347
  - grant of copyright license, 345
  - grant of patent license, 345
  - limitation of liability, 347
  - redistribution provisions, 345–346
  - submission of contributions, 346
  - trademarks, 346
- Apache Module Developers mailing list, 16
- #apache-modules chat channel, 17
- Apache Portable Runtime (APR), 53–54
  - apr\_status\_t and return values, 58
  - APU library, 56–57
  - buckets and brigades, 74–75
  - conditional compilation, 59

**Apache Portable Runtime (APR), *continued***

- data structures, 70–73
- databases, 79–83
- declaration macros, 58
- developers list for, 16
- encoding and cryptography, 76–77
- extensions, 79
- filesystems, 76
- internationalization, 69
- libraries, 21
- modules, 54–55
- namespacing, 57–58
- networks, 76
- pools, 29, 78–79
  - basic memory management*, 61–62
  - connection*, 290–292
  - generalized memory management*, 62–63
  - implicit and explicit cleanup*, 63–64
  - lifetime*, 65–67
  - limitations*, 68
  - subpools*, 98
- processes and threads, 78
- reference manual, 57
- resource management, 59
  - lifetime*, 65–67
  - pool limitations*, 68
  - pools*, 61–64
  - problem of*, 60–61
- strings and formats, 69
- summary, 83–84
- time and date, 70
- URI handling, 77

**Apache Software Foundation (ASF), 3–6****ApacheCon conferences, 17****APLOG\_DEBUG level, 326****APLOG\_INFO level, 326–327****APLOG\_MARK macro, 325****APLOG\_TOCLIENT level, 325****Appendices in HTTP specification**

- additional features, 524–525
- HTTP entities vs. RFC 2045 entities, 521–524

## Internet media types

- message/http and application/http*, 518–519
- multipart/byteranges*, 519–520

- tolerant applications, 520–521

**application/http type, 518–519****application/x-www-form-urlencoded format, 138****Applications, tolerant, 520–521****Applications development**

- Apache history, 1–3
- Apache Software Foundation, 3–6
- codebase, 7–9
- developers, 10–11
- forums, 9–10
- further reading, 16–19
- intellectual property in, 12–16
- participation in forums, 11
- process, 6
- summary, 19

**APR. *See* Apache Portable Runtime (APR)****apr\_ header files, 39****apr\_allocator function, 106****apr\_array\_header\_t type, 70****#apr chat channel, 17****apr\_conf\_vector\_t type, 29****apr\_date module, 70****apr\_dbd API, 82–83, 290, 292–294**

- database operations in, 294–298
- functions in, 298–301

**apr\_dbd\_check\_conn function, 299****apr\_dbd\_close function, 298****apr\_dbd\_driver\_t type, 294, 307–308****apr\_dbd\_error function, 298, 301****apr\_dbd\_escape function, 301****apr\_dbd\_get\_driver function, 298****apr\_dbd\_get\_entry function, 300****apr\_dbd\_get\_name function, 300****apr\_dbd\_get\_row function, 297, 300****apr\_dbd.h file, 298****apr\_dbd\_init function, 298****apr\_dbd\_internal.h file, 307–308****apr\_dbd\_name function, 298**

- `apr_dbd_native` function, 293
- `apr_dbd_native_handle` function, 298
- `apr_dbd_num_cols` function, 300
- `apr_dbd_num_tuples` function, 300
- `apr_dbd_open` function, 298
- `apr_dbd_pquery` function, 295, 301
- `apr_dbd_prepare` function, 296, 301
- `apr_dbd_prepared_t` type, 294, 308
- `apr_dbd_pselect` function, 295, 301
- `apr_dbd_pvquery` function, 295, 301
- `apr_dbd_pvselect` function, 295, 301
- `apr_dbd_query` function, 295, 299
- `apr_dbd_results_t` type, 294
- `apr_dbd_row_t` type, 294, 308
- `apr_dbd_select` function, 295–296, 300
- `apr_dbd_set_dbname` function, 299
- `apr_dbd_t` type, 294, 308
- `APR_DBD_TRANSACTION_COMMIT` mode, 297
- `apr_dbd_transaction_end` function, 299
- `APR_DBD_TRANSACTION_IGNORE_ERR` ORS mode, 297
- `apr_dbd_transaction_mode_get` function, 299
- `apr_dbd_transaction_mode_set` function, 299
- `APR_DBD_TRANSACTION_ROLLBACK` mode, 297
- `apr_dbd_transaction_start` function, 299
- `apr_dbd_transaction_t` type, 294, 308
- `apr_dbm.h` file, 80
- `apr_dbm` module, 80–81, 106
- `APR_DECLARE_EXTERNAL_HOOK` macro, 270
- `APR_DECLARE` macro, 58, 284
- `apr_file_info` module, 76
- `apr_file_io` module, 76
- `apr_filepath` function, 111
- `apr_fnmatch` module, 76
- `apr_global_mutex` function, 78, 102
- `APR_HASH_KEY_STRING` macro, 73
- `apr_hash` module, 70
- `apr_hash_t` type, 72
- `APR_HOOK_FIRST` macro, 171, 269
- `APR_HOOK_LAST` macro, 269
- `APR_HOOK_MIDDLE` macro, 269
- `APR_HOOK_REALLY_FIRST` macro, 269
- `APR_HOOK_REALLY_LAST` macro, 269
- `apr_hooks.h` file, 268, 270
- `apr_hooks` module, 79
- `apr_iconv` library, 69
- `APR_IMPLEMENT_EXTERNAL_HOOK_R` UN\_ALL macro, 271
- `APR_IMPLEMENT_OPTIONAL_HOOK_R` UN\_ALL macro, 270
- `apr_memcache` module, 106
- `apr_mmap` module, 76
- `apr_network_io` module, 76
- `apr_off_t` type, 231
- `apr_optional_hooks.h` file, 268, 270
- `apr_optional_hooks` module, 79
- `apr_optional` module, 79
- `apr_palloc` function, 286
- `apr_poll` module, 76
- `apr_pool_cleanup` function, 280
- `apr_pool_cleanup_kill` function, 63–64
- `apr_pool_cleanup_register` function, 64, 283
- `apr_pool_cleanup_run` function, 64
- `apr_pool_clear` function, 99
- `apr_pool_t` type, 29
- `apr_proc_mutex` function, 78, 102
- `apr_queue_create` function, 73
- `apr_queue` module, 70
- `apr_queue_t` type, 73
- `apr_read_type_e`, 231
- `apr_relist_create` function, 278
- `apr_relist_destroy` function, 278
- `apr_relist` module, 78
  - implementing, 278–284
  - for resource reuse, 99
- `apr_ring` module, 70, 73
- `apr_rmm` module, 105–106
- `apr_shm` function, 78, 104–105
- `apr_signal` function, 78

**apr\_status\_t** type, 58, 165–166, 298  
**apr\_strings** module, 69  
**apr\_strmatch** module, 69  
**apr\_table\_addn** function, 71  
**apr\_table\_merge** function, 72  
**apr\_table\_mergen** function, 72  
**apr\_table** module, 70  
**apr\_table\_overlap** function, 72  
**apr\_table\_set** function, 72  
**apr\_table\_setn** function, 71–72  
**apr\_table\_t** type, 71  
**apr\_thread\_cond** function, 78  
**apr\_thread\_mutex** function, 78  
**apr\_thread\_proc** function, 78  
**apr\_thread\_rwlock** function, 78  
**apr\_time\_from\_sec** macro, 70  
**apr\_time** module, 70  
**apr\_time\_sec** macro, 70  
**apr\_time\_t** type, 70  
**apr\_uri** module, 77  
**apr\_uri\_parse** function, 77  
**apr\_uri\_t** type, 77  
**apr\_uri\_unparse** function, 77  
**APR-UTIL (APU) library**, 56–57  
**apr\_xlate** module, 69  
**APU\_MODULE\_DECLARE\_DATA** module, 307  
**apvfs** library, 76  
**apxs** utility, 128  
**Architecture**. *See* Platform and architecture  
**ARRAY\_merge** function, 254–255  
**Arrays**, 70–71  
**Articles**, websites for, 19  
**ASF (Apache Software Foundation)**, 3–6  
**ASF members**, 5  
**#asfinfra** chat channel, 17  
**Asterisks (\*)** in HTTP rules, 367  
**Attribute/value pairs** in transfer codings, 376  
**Audits for intellectual property violations**, 16  
**Augmented BNF**, 366–368  
**auth\_checker** hook, 45, 155–156  
**AuthDBDUserPWQuery**, 304

## Authentication

**AAA**. *See* AAA (Access, Authentication, and Authorization)  
 in Apache 2.1/2.2, 183–184  
 basic, 178–179  
 cookies for, 198–199  
 credentials, 509  
 digest, 179, 193–195  
 HTTP/1.1 specification, 424  
 methods, 187  
 in **mod\_authn\_dbd**, 304  
 providers, 188–190, 193–195  
 Require directive, 186  
 Web login, 180  
**Authentication dialog**, 179–180  
**authn\_dbd\_password** function, 305  
**authn\_dbd\_prepare** function, 304–305  
**authn\_provider** structure, 194  
**Author addresses in HTTP/1.1 specification**, 516–518  
**Authorization**  
 AAA. *See* AAA (Access, Authentication, and Authorization)  
 in Apache 2.1/2.2, 184–185  
**Authorization field**, 460–461  
**authz\_dbd\_login** function, 270–271

## B

**Backports**, 8  
**Backslash characters (\)** in HTTP, 369  
**Backus-Naur Form (BNF)**, 366–368  
**Bad Gateway status code**, 423  
**Bad Request status code**, 418  
**Bars (|)** in HTTP rules, 366  
**Basic authentication**, 178–179  
**Basic authentication providers**, 188–190, 193–195  
**Basic rules in HTTP**, 368  
**Binary exponential backoff algorithm**, 402–403

- block argument for filters, 208**
  - Blocking for input filters, 231**
  - Blocks, coding, 86–87**
  - BNF (Backus-Naur Form), 366–368**
  - Bodies**
    - entities, 395
    - HTTP messages, 385–386
  - Braces ({} for blocks, 86**
  - Branches in code repository, 7**
  - Breakpoints, 329–330**
  - Brigades, 74–75, 132–137, 210–211, 217–227**
  - Broken connections, 138 debugging, 339–340**
  - Browsers**
    - authentication dialogs for, 197–199
    - delays and broken connections in, 339–340
  - Buckets, 74–75**
    - for handlers 132–137**
    - for filters, 217–227
    - metadata, 205
    - for rings, 73
  - Buffer overflows, 109**
  - Bugzilla database, 10**
  - Builds**
    - cross-platform, 284–288
    - forum participation for, 11
  - Byte ranges**
    - for cache responses, 448
    - working with, 492–493
  - in apache, 203**
- C**
- C language**
    - stubs, 122
    - wrappers, 63
  - C++ language**
    - for pools, 64
    - throw/catch structures, 116
  - Cache-Control general-header field, 431, 461–462**
    - basic expiration mechanism for, 464–466
    - cacheable responses for, 463–464
    - extensions, 470
    - revalidation and reload controls for, 466–469
  - Cacheable responses in HTTP/1.1 specification, 362**
  - Caches**
    - in HTTP, 362, 427–428
      - client-controlled behavior for, 432*
      - control mechanisms for, 431*
      - correctness of, 429*
      - errors in, 450*
      - expiration model, 433–438*
      - history lists, 452–453*
      - invalidation in, 451–452*
      - for negotiated responses, 449–450*
      - replacement of, 452*
      - response cacheability, 444–445*
      - responses from, 445–448*
      - security considerations, 509–510*
      - shared and non-shared, 450*
      - side effects in, 451*
      - validation model, 438–444*
      - warnings for, 430–432*
      - write-through mandatory, 452*
    - for pools, 96
    - in request processing, 160
    - for shared memory, 104
  - Calculations in cache expiration model, 434–437**
  - Callback functions**
    - for filters, 205
    - for modules, 125–126
  - Canonical form**
    - conversion to, 522
    - media types, 379–380
  - case statements, 87**
  - CGI (Common Gateway Interface), 123, 291**
  - Character sets**
    - in HTTP/1.1 specification, 374–375
    - in RFC 2068, 528
  - Chat for developers, 9, 17**
  - check\_password function, 194**

- check\_user\_id hook**
  - for authentication, 183–184, 190
  - purpose, 45
  - in request processing, 155–156
- child\_init function**
  - for garbage collection, 96
  - for reslist, 281
  - for threads, 93–94, 117
- Child pools, 67**
- chroot command, 113**
- Chunked transfer coding, 204, 377–378**
- CLA (Contributor License Agreement), 15**
  - corporate, 353–356
  - individual, 349–353
- Classic LAMP Model, 291**
- Cleanup, pool, 63–64**
- client\_cookie function, 271**
- Client error 4xx status code definitions, 418–423**
- client\_login function, 270**
- Clients and client behavior**
  - for caches, 432
  - in HTTP/1.1 specification, 361
  - idle, 509
  - for prematurely closed connections, 402–403
- Clockless origin server operation, 479–480**
- cmd\_parms\_struct structure, 243**
- Codebase, 7**
- Coding, 85**
  - blocks, 86–87
  - comments, 87–88
  - declarations, 87
  - flow control, 87
  - forum participation for, 11
  - functions, 86
  - lines, 86
- Combining cache response headers, 447–448**
- <Comment> container, 258**
- Comments**
  - coding, 87–88
  - in containers, 25
  - HTTP, 368–369
- COMMIT for transactions, 297**
- Commit-Then-Review (CTR) code, 8**
- Committers in Apache Software Foundation, 5**
- Common Gateway Interface (CGI), 123, 291**
- Communicating between modules, 90–92**
- Comparisons, URI, 372**
- Compatibility of HTTP versions, 525–529**
- Compilation**
  - conditional, 59
  - module, 128
- Complex parsing, filters for, 221–225**
- Complexity in handler vs. filter decisions, 47**
- compress encoding format, 376**
- Compression and decompression, 49, 203, 232–235**
- Conditional compilation, 59, 284–288**
- Conferences for developers, 17**
- Configuration 237–262**
  - AAA modules, 193
  - basics, 41–42
  - filters, 213–215
  - modules. *See* Modules
  - pools, 66
  - in start-up phase, 23–25
- Configuration data**
  - scope of, 89
  - in thread-safe programming, 92
- Configuration records, 29**
- Configuration vectors, 88**
- Conflict status code, 420–421**
- Conflicts**
  - directives, 238
  - in httpd.conf, 251–253
- conn\_rec object, 29**
  - definition, 37–38
  - for filters, 203
- CONNECT method, 410**
- Connection general-header field, 470–471**
- Connections**
  - debugging, 339–340
  - filters, 203–204
  - HTTP/1.0 compatibility with, 526–527



- HTTP/1.1, 359
  - message transmission requirements, 400–403
  - persistent, 396–400, 526–527
  - pooling, 67, 290–292
  - scope of, 90
- Constructor/destructor model, 60**
- Containers, 24–25**
  - context checking, 255–257
  - custom, 257–261
  - merging, 251–254
- Content-Base in RFC 2068, 527**
- Content codings in HTTP/1.1 specification, 375–376**
- Content-Disposition response-header field**
  - for default filenames, 524–525
  - security issues, 509
- Content-Encoding entity-header field, 395–396, 471–472, 523**
- Content filters, 202–205**
- Content generators, 123**
  - default handlers, 144–148
  - HelloWorld module. *See* HelloWorld module
  - reading form data, 138–144
  - in request processing, 42–43, 48
  - summary, 148
- Content-Language entity-header field, 382, 472–473**
- Content-Length entity-header field, 216, 387, 473**
- Content-Location entity-header field, 473–474, 489**
- Content-MD5 entity-header field, 216, 474–476**
- Content negotiation, 424–425**
  - agent-driven, 426–427
  - HTTP/1.1, 360
  - in apache, 158–160
  - server-driven, 425–426
  - transparent, 427
- Content-Range entity-header field, 383, 476–478**
- Content-Transfer-Encoding (CTE) field, 523**
- Content-Type entity-header field, 395–396, 478**
  - See also* `ap_set_content_type`
- Context in configuration, 25, 255–257**
- Continuation lines, 86**
- Continue status code, 400–402, 410–411**
- Contributor License Agreement (CLA), 15**
  - corporate, 353–356
  - individual, 349–353
- Control mechanisms for caches, 431**
- Conversion**
  - to canonical form, 522
  - of date formats, 522
- Cookies for authentication, 198–199**
- copy function for buckets, 74**
- Copyleft, 13**
- Copyright license, 345**
- copyright notice in HTTP1.1 specification, 358**
- Copyright statement in HTTP/1.1 specification, 530**
- Core dumps, debugging, 332–333**
- CoredumpDirectory directive, 332**
- Corporate CLA, 353–356**
- Costs, license, 12**
- CR LF**
  - HTTP, 368–369
  - with media types, 379–380
- Crashes, tracing, 331–332**
- CREATE keyword in RFC 2068, 527**
- Credentials, authentication, 509**
- Cross-MPM programming, 101–102**
  - process and global locks in, 102–104
  - shared memory in, 104–106
- Cross-platform issues**
  - API builds, 284–288
  - programming, 99–101
- Cryptography, 76–77**
- CTR (Commit-Then-Review) code, 8**
- Cursors, 296–297**
- Custom containers, 257–261**
- Custom login schemes, 195–199**

**D****Data axis for filters, 46–49****Data compression and decompression, 49****Data structures, 70**

arrays, 70–71

hash tables, 72–73

for module configuration, 239

queues, 73

rings, 73

tables, 71–72

**Data types, 294–296****Data vs. metadata, 153****Database framework, 79–80, 289**

ap\_dbd, 302–303

Apache 1.x/2.0 vs. Apache 2.2, 290

apr\_dbd, 82–83, 292–294

*functions in, 298–301**operations in, 294–298*

apr\_dbm, 80–81

connection pooling, 290–292

DBD architecture, 292

DBD driver, 306–320

mod\_authn\_dbd application, 303–306

summary, 320

**Date**

format conversions, 522

in HTTP/1.1 specification, 373–374

module for, 70

**Date general-header field, 478–480****Day-based anonymous authentication, 193****DBD, drivers in, 292, 306–307**

apr\_dbd\_internal.h for, 307

exporting, 307–308

functions, 309–320

**dbd\_close function, 280****dbd\_construct function, 278–279****dbd\_destruct function, 280****dbd\_mysql driver, 311****dbd\_setup function, 280–281****DDoS (distributed denial of service) attacks, 109****Debugging, 323**

core dumps, 332–333

crash tracing, 331–332

filters, 338–341

hooks for, 336–338

logging for, 324–327

modules for, 333–337

MPMs for, 331

running under debugger, 327–333

servers, 329–331

summary, 341

help for system administrators, 326–327

**Declarations**

coding, 87

macros for, 58

module APIs, 286–288

**DECLINED handler value, 126****default\_handler function, 144****Defaults**

handlers, 144–148

text media type, 379–380

**deflate encoding format, 376****deflate\_in\_filter function, 232–235****Delays, debugging, 339–340****DELETE method**

cache invalidation with, 451

as idempotent method, 404

working with, 409

**Deletions, cache invalidation after, 451–452****Delta seconds, 374****Denial of service (DoS) attacks**

protecting against, 109–110

on proxies, 510

**Deny directive, 182****Deny from directive, 186****Denying access, 186–187****Dependencies, external. See Libraries****destroy function, 74****Developers, 10–11**

in Apache Software Foundation, 5

documentation for, 18

mailing list, 16

**Development branches in code repository, 7****Development forums, 9–10****Digest authentication**

- overview, 179
- providers, 193–195

**Directives**

- Cache-Control field, 461–462
- characteristics, 24
- context for, 25
- for filters, 211
- functions for, 242–246
- for module configuration, 238, 242–250
- preprocessor, 285–286
- user data in, 244–245

**Directories, configuring, 240–241****<Directory> container, 238, 240, 251–252****<DirectoryMatch> directive, 241****Disambiguating in cache expiration model**

- expiration values, 437
- multiple responses, 437–438

**Disconnected operation warn code, 503****Distributed denial of service (DDoS) attacks, 109****Diverting requests, 161–163****DNS spoofing, 508****do\_garbage function, 97–98****Document variants, 168–171****Documentation**

- APR, 57
- forum participation for, 11
- websites for, 18

**DoS (denial of service) attacks**

- protecting against, 109–110
- on proxies, 510

**doxygen format, 57****Drivers, DBD, 292, 306–307**

- apr\_dbd\_internal.h for, 307
- dbd\_mysql, 311
- exporting, 307–308
- functions, 309–320

**E****Encoding**

- APR, 76–77
- HTTP formats, 375–376
- in form data, 138–139
- in URIs, 506

**enctype attribute, 138****End-to-end headers, 445–446****End-to-end reload, 467****Entities**

- filtering, 204
- HTTP vs. RFC 2045, 521–524
- in HTTP/1.1 specification, 360, 394
  - bodies*, 395–396
  - header fields*, 395
  - length*, 396

**Entity tags**

- for cache validation, 439, 442–444
- in HTTP/1.1 specification, 382–383

**Environment variables**

- CGI, 41
- for filters, 214–215
- in request\_rec, 30
- setting, 24, 45, 55, 91

**EOS bucket type, 226**

- checking for, 219
- in debugging, 340
- support for, 75

**err\_headers\_out table, 92****Error log, 324–326****ErrorDocument directive, 163****Errors and error documents**

- in caches, 450
- connection monitoring for, 400
- I/O, 137–138
- in modules, 172–174
- in request processing, 162–163
- in secure programming, 109

**ETag response-header field, 216, 480****Event MPMs, 27**

**exec command**, 113  
**EXEC\_ON\_READ option**, 246  
**Existing parsers, filtering through**, 225–227  
**Expanded macros**, 121  
**expat library**, 206, 225–226  
**Expect request-header field**, 480  
**Expectation Failed status code**, 423  
**Expiration and expiration models**, 362–363
 

- age calculations, 434–436
- for Cache-Control general-header field, 464–466
- disambiguating, 437–438
- expiration calculations, 437
- heuristic, 433–434
- server-specified, 433

**Expires entity-header field**, 481–482  
**Explicit cleanup of pools**, 63–64  
**Explicit expiration time in HTTP/1.1 specification**, 362  
**Explicit user agent warnings**, 431–432  
**Exporting**

- drivers, 307–308
- functions, 264
- optional hooks, 270–271

**Extending API**, 263–264
 

- APR, 79
- cross-platform builds, 284–288
- hooks and optional hooks, 267–271
- new function implementation, 264–266
- provider API, 272–277
- providing services, 277–284
- summary, 288

**External dependencies. See Libraries**

## F

**Fastcgi**, 28  
**fatal\_exception hook**, 268  
**Fatal exception hooks**, 268, 336  
**Feasibility factor in handler vs. filter decisions**, 46

**fetch\_dbm\_value function**, 81  
**File bucket type**, 75  
**File names, attacks based on**, 507–508  
**<Files> container**, 238, 240, 251  
**Filesystems**

- APR, 76
- mapping, 156–158
- security in, 111–113

**filter\_func function**, 210  
**filter\_hooks function**, 214  
**filter\_init\_func function**, 210  
**filter\_insert function**, 213–214  
**filter-oriented output**, 133  
**FilterChain directive**, 213  
**Filters and filter modules**, 201–202
 

- buckets for, 217–221
- callback functions for, 205
- for complex parsing, 221–225
- content, protocol, and connection, 202–205
- debugging, 338–341
- through existing parsers, 225–227
- input, 202, 207–208, 210–211, 230–235
- mod\_filter, 213
- objects, 208–210
- output, 202, 207, 210–211
- pipelining for, 205–206
- for pools, 67
- postprocessing and preprocessing, 212
- for protocol handling, 215–217
- request processing, 46–49
- self-configuration, 213–215
- stdio-like I/O, 227–230
- summary, 235–236

**find\_file function**, 112–113  
**First-hand responses in HTTP/1.1 specification**, 362  
**Fixups hook**

- purpose, 45
- in request processing, 155–156

**Fixups phase in subrequests**, 166  
**#flastmod SSI element**, 165  
**Flexibility with libraries**, 119

**Flow and flow control**

- coding, 87
- for message connections, 400
- upstream/downstream, 363

**Flush bucket type, 75****Forbidden status code, 186, 419****Form data, parsing, 138–144****form\_data function, 265****form\_hooks function, 266****Form-processing modules, 49****form\_value function, 142, 265****Format strings, 295–296****Formats**

- APR, 69
- date/time, 373–374, 522

**Found status code, 415–416****Free Software Foundation (FSF), 13****Free software licenses, 12****Fresh responses in HTTP/1.1 specification, 363****Freshness lifetime in HTTP/1.1 specification, 363****From request-header field, 482****FSF (Free Software Foundation), 13****#fsize SSI element, 165****Full copyright statement in HTTP/1.1 specification, 530****Full date formats in HTTP/1.1 specification, 373–374****Fully generic shared memory, 106****Functions**

- apr\_dbd API, 298–301
- callback, 125–126, 205
- coding, 86
- DBD driver, 309–320
- for directives, 242–246
- exporting, 264
- implementing, 264–266
- optional, 265–266
- registering, 115
- static, 129
- using, 266

**G****Garbage collection**

- function for, 96–98
- in resource management, 60–61

**Gateway Timeout status code, 424****Gateways in HTTP/1.1 specification, 362****GDBM, 80–81****General header fields, 387–388****Generic grammar in HTTP/1.1 specification, 366–369****GET method**

- as idempotent method, 404
- OK status code with, 411
- side effects of, 451
- working with, 405–406

**Global data in thread-safe programming, 92****Global locks, 102–104****global mutexes, 102–103****GLOBAL\_ONLY macro, 255****global\_score entry, 334****Global variables, 117, 338****GNU General Public License (GPL), 13–14****Gone status code, 421****GPL (GNU General Public License), 13–14****Grammar in HTTP/1.1 specification, 366–369****Grant of copyright license, 345****Grant of patent license, 345****gzip encoding format, 375****H****Handler field, 127****Handler hook**

- purpose, 45
- in request processing, 156

**Handlers**

- default, 144–148
- vs. filters, 46–48
- pools, 66

**Hash characters (#)**

- for comments, 25
- HTTP rules, 367

**Hash tables, 72–73****HEAD method, 404**

- as idempotent method, 404
- OK status code with, 411
- side effects of, 451
- working with, 406

**Header field definitions in HTTP/1.1 specification, 453**

- Accept, 453–455
- Accept-Charset, 455–456
- Accept-Encoding, 456–457
- Accept-Language, 457–458
- Accept-Ranges, 459
- Age, 459
- Allow, 459–460
- Authorization, 460–461
- Cache-Control, 461–470
- Connection, 470–471
- Content-Encoding, 471–472
- Content-Language, 472–473
- Content-Length, 473
- Content-Location, 473–474
- Content-MD5, 474–476
- Content-Range, 476–478
- Content-Type, 478
- Date, 478–480
- ETag, 480
- Expect, 480–481
- Expires, 481–482
- From, 482
- Host, 482–483
- If-Match, 483–484
- If-Modified-Since, 484–486
- If-None-Match, 486–487
- If-Range, 487–488
- If-Unmodified-Since, 488
- Last-Modified, 488–489
- Location, 489
- Max-Forwards, 489–490
- Pragma, 490–491

- Proxy-Authenticate, 491
- Proxy-Authorization, 491
- Range, 492–494
- Referer, 494
- Retry-After, 494–495
- Server, 495
- TE, 495–496
- Trailer, 497
- Transfer-Encoding, 497
- Upgrade, 498–499
- User-Agent, 499
- Vary, 499–500
- Via, 500–501
- Warning, 501–504
- WWW-Authenticate, 504

**header\_parser hook, 197**

- purpose, 45
- in request processing, 155–156

**Headers and header fields 153–155**

- for cache responses, 445–448
- contents, 384–385
- entities, 395
- filter, 204
- general, 387–388
- in HTTP/1.1. *See* Header field definitions in HTTP/1.1 specification
- Range, 493–494
- request messages, 154, 391
- response messages, 394

**headers\_in table, 91****headers\_out table, 92****Heap bucket type, 75****HelloWorld module, 124**

- completed, 127–129
- handler field, 127
- I/O, 132–133
  - errors*, 137–138
  - input*, 134–137
  - output*, 133–134
- request\_rec for, 129–130
- response page for, 130–132
- return values, 126–127
- skeleton, 124–126

**Heuristic expiration**

- in cache expiration model, 433–434
- in HTTP/1.1 specification, 363

**Heuristic expiration warn code, 503****Hexadecimal numeric characters, 369****History lists, 452–453****Hooks, 267**

- analysis of, 267–269
- for debugging, 336–338
- optional, 270–271
- order of execution, 269
- pools, 66
- processing, 44–46, 50–51
- in request processing, 155–156

**Hop-by-hop headers, 445–446****Host access**

- in AAA, 181
- in Apache 2.1/2.2, 183

**Host request-header field, 390, 482–483, 526****.htaccess file**

- for authentication, 186
- purpose, 238–239
- for request processing security, 160

**http\_header files, 39****HTTP/1.0, changes from, 525–526****HTTP/1.1 specification**

- abstract, 358
- access authentication, 424
- acknowledgments, 510–512, 530
- appendices, 518–529
- author addresses, 516–518
- caching. *See* Caches
- character sets, 374–375
- connections
  - message transmission requirements, 400–403*
  - persistent, 396–399*
- content codings, 375–376
- content negotiation, 424–427
- copyright notice, 358
- date/time formats, 373–374
- entities, 394–396
- entity tags, 382–383

full copyright statement, 530

header fields. *See* Header field definitions in  
HTTP/1.1 specification

index, 529

language tags, 382

media types, 379–381

memo status, 357

method definitions, 403–410

notational conventions and generic grammar,  
366–369

overall operation, 364–366

product tokens, 381

purpose, 358–359

quality values, 381–382

range units, 383

references, 512–516

requirements, 359

security considerations, 504–510

status codes. *See* Status code definitions

terminology, 359–365

transfer codings, 376–378

Uniform Resource Identifiers, 371–372

versions, 370–371

**http\_config.h file, 40, 246–247, 255****http\_connection.h file, 40****http\_core.h file, 40****HTTP entities vs. RFC 2045 entities, 521–524****http\_log.h file, 40, 324****http\_main.h file, 40****HTTP messages**

bodies, 385–386

connection transmission requirements,  
400–403

headers, 154, 384–385, 387–388, 391

HTTP protocol for, 152–153

length, 386–387

request. *See* Request processing and messages

response, 153, 392–394

types, 383–384

**http\_protocol.h file, 40, 132, 134****http\_request.h file, 40****http URL, 372**

**HTTP Version Not Supported status code, 424**

**http\_vhost.h file, 40**

**httpd.conf file**

**intro 23-25**

for authentication, 186

conflicts in, 251-253

containers in, 24-25

*context checking, 255-257*

*custom, 257-261*

*merging, 251-254*

for core dumps, 332

directives. *See* Directives

for request processing security, 160

**#httpd-dev chat channel, 17**

**httpd.h file**

conn\_rec in, 37

contents, 40

request\_rec in, 30-35

server\_rec in, 35-37

**Hypertext Transfer Protocol (HTTP). *See***

**HTTP messages; Request processing  
and messages**

## I

**IANA (Internet Assigned Numbers Authority)**

Character Set registry, 374

for content codings, 375

for transfer codings, 377

**Idempotent methods, 404**

**identity encoding format, 376**

**Idle clients, 509**

**If-Match request-header field, 483-484**

**If-Modified-Since request-header field,  
484-486**

**If-None-Match request-header field, 486-487**

**If-Range request-header field, 487-488**

**If-Unmodified-Since request-header field, 488**

**Image processing filters, 49**

**Immortal bucket type, 75**

**Implementing new functions, 264-266**

**Implicit cleanup of pools, 63-64**

**Implied linear white space, 368-369**

**Inbound/outbound paths, 364**

**#include file SSI command, 165-166**

**#include virtual SSI command, 165-166**

**Incomplete responses in caches, 450**

**Index in HTTP/1.1 specification, 529**

**Individual CLA, 349-353**

**Information gathering in request processing,  
163-167**

**Informational 1xx status code definitions,  
410-411**

**Initialization**

library, 116-117

pool, 66

server start-up, 22-23, 268

**Input filters, 202, 230**

blocking, 231

callback functions for, 205, 207-208

example, 232-235

modes for, 231

readbytes for, 231-232

strategies for, 210-211

**Inputs**

module, 134-137

safe use of, 108

validating, 107-108

**insert\_filter hook**

in request processing, 156

for self-configuration, 213-214

**Intellectual property, 12-16**

**Interactive online forums, 16-17**

**Internal redirects in request processing,  
161-162**

**Internal Server Error status code, 423**

**Internationalization, 69**

**Internet Assigned Numbers Authority (IANA)**

Character Set registry, 374

for content codings, 375

for transfer codings, 377

**Internet media types**

in HTTP/1.1 specification, 379-381

message/http and application/http, 518-519

multipart/byteranges, 519-520



**Internet Relay Chat (IRC), 9, 17**

**Invalidation, cache, 451–452**

**invoke\_cmd function, 330**

**I/O**

filter, 230–235

*stdio-like, 227–230*

*strategies, 210–211*

module, 132–133

*errors, 137–138*

*input, 134–137*

*output, 133–134*

*reading form data, 138–144*

**iovec type, 75**

**IP addresses, conserving, 526**

**IRC (Internet Relay Chat), 9, 17**

## K–L

**Key/value pairs for form data, 138**

**Keys, hash table, 72–73**

**Labels, 295–296**

**LAMP architecture, 289–292**

**Language tags, 382**

**Last-modified dates in cache validation, 439, 442–444**

**Last-Modified entity-header field, 485, 488–489**

**lb\_scores entry, 335**

**Length of HTTP messages, 386–387**

**Length Required status code, 421**

**Levels**

authentication, 178–180

logging, 325–327

**libdbi, 292**

**libjpeg library, 114–115**

**libmysqlclient library, 116**

**Libraries**

APR, 21

apvfs, 76

with filters, 206

good practice, 114–118

module building with, 118–120

serf, 75

state changes in, 117–118

third-party, 114

**libxml2 library, 117–118, 206**

**Licenses**

Apache, 12–14, 343–348

corporate CLA, 353–356

GPL, 13

individual CLA, 349–353

**Lifetime**

in module data management, 88–90

in resource management, 65–67

**<Limit> container, 256–257**

**<LimitExcept> container, 256**

**Line-length limitations in MIME, 524**

**Linear white space (LWS), 368–369, 384–385**

**Lines**

coding, 86

in HTTP requests, 154

**Linux in LAMP architecture, 289**

**Literals, 366**

**load\_module function, 330–333**

**LoadFile directive, 118–120**

**LoadModule directive, 119**

**<Location> container, 238, 240–241, 251–252**

**Location header spoofing, 508–509**

**Location response-header field, 489**

**Locks in cross-MPM programming, 102–104**

**log\_transaction hook, 45**

**logger hooks, 156**

**Logging and log information**

abuse of, 505

for debugging, 324–327

in request processing, 156, 161

**Login**

custom schemes, 195–199

Web, 180

**LogLevel directive, 325**

**longjmp function, 114**

**LWS (linear white space), 368–369, 384–385**

**M****<Macro> container, 259–260****macro\_section function, 260****macro\_t type, 259****Macros**

APR, 58

expanded, 121

**Mailing lists, 9, 16****make\_label function, 296****Malformed requests, 163****Malicious requests, 163****malloc function, 68****map\_to\_storage hook**

purpose, 44

in request processing, 155–156

**Mapping in HTTP requests, 154, 156–158****Markup parsing modules, 49****max-age directive, 465, 467–468****Max-Forwards request-header field, 489–490****max-stale directive, 466****MaxRequestsPerChild, 96, 98****MD5 hash, 194****Media types**

in HTTP/1.1 specification, 379–381

message/http and application/http, 518–519

multipart/byteranges, 519–520

**Memory**

managing, 61–63, 96

shared, 104–106

**merge\_config function, 252****Merging containers, 251–254****Meritocracy in Apache Software Foundation, 4****message/http type, 518–519****Messages, HTTP. *See* HTTP messages****Metadata**

vs. data, 153

private, 160

**Metadata buckets, 205****Method definitions, 403**

CONNECT, 410

DELETE, 409

GET, 405–406

HEAD, 406

idempotent, 404

OPTIONS, 404–405

POST, 407

PUT, 408–409

safe, 403–404

TRACE, 409

**Method Not Allowed status code, 419****Method tokens, 388****Metux MPM, 28****MHTML messages, 524****MIME character set, 374**

in HTTP/1.1 specification, 380–381

line-length limitations, 524

**MIME-Version header field, 521–522****min-fresh directive, 466****Miscellaneous persistent warning warn code, 504****Miscellaneous warning warn code, 503****Missing character sets, 375****Mmap bucket type, 75****mod\_header files, 39****mod\_access module, 182****mod\_alias module, 43, 157****mod\_auth\_basic module, 184, 187, 190, 272****mod\_auth\_cookie module, 92****mod\_auth\_dbm module, 182****mod\_auth\_digest module, 182, 184, 187, 272****mod\_authn\_alias module, 184****mod\_authn\_anon module, 184, 272****mod\_authn\_day module, 194****mod\_authn\_dbd module, 184, 196, 272, 303–306****mod\_authn\_dbm module, 184, 272****mod\_authn\_default module, 184****mod\_authn\_file module, 184, 272****mod\_authn\_ldap module, 272****mod\_authnz\_day module, 194****mod\_authnz\_ldap module, 184–185, 265****mod\_authz\_dbd.h file, 270****mod\_authz\_dbd module, 185, 196–197, 270–271**

- mod\_authz\_dbm** module, 185
- mod\_authz\_default** module, 185
- mod\_authz\_groupfile** module, 269
- mod\_authz\_host** module, 183–184
- mod\_authz\_owner** module, 185, 269
- mod\_authz\_user** module, 185, 191–192
- mod\_backdoor** module, 333, 337
- mod\_backtrace** module, 333, 336
- mod\_cband** module, 110
- mod\_dbd.h** file, 41, 285–286
- mod\_dbd** module, 277–278
- mod\_deflate** module, 49, 214, 232, 261
- mod\_diagnostics** module, 333, 338–341
- mod\_env** module, 91
- mod\_evasive** module, 110
- mod\_expat** module, 227
- mod\_ext\_filter**, 48
- mod\_fcgid**, 28
- mod\_filter** module, 213–214, 216
- mod\_form.h** file, 266
- mod\_form** module, 265
- mod\_headers** table, 91
- mod\_helloworld.c** file, 127
- mod\_include** module, 165, 264
- mod\_includes** filter, 48
- mod\_info** module, 333
- mod\_ldap** module, 105–106
- mod\_line\_edit** module, 222
- mod\_load\_average** module, 110
- mod\_macro** module, 259
- mod\_negotiation** module, 43, 158–160
- mod\_proxy** module, 248
- mod\_proxy\_html** module, 225, 248–249, 326–327, 340–341
- mod\_publisher** module, 264, 273
- mod\_rewrite**, 43, 91, 326
- mod\_robots** module, 110
- mod\_ruid**, 28
- mod\_security** module, 110
- mod\_setenvif** module, 91
- mod\_ssl** filter, 49
- mod\_status** module, 333–334
- mod\_transform.h** file, 264
- mod\_transform** module, 168, 264
  - mod\_transform\_set\_XSLT** function, 264, 284
  - mod\_transform\_XSLTDoc** function, 264
- mod\_txt** module, 217–221
  - configuration for, 244–245
- mod\_upload** module, 140
- mod\_watchdog** module, 333
- mod\_whatkilledus** module, 333, 336
- mod\_xhtml** module, 276
- mod\_xmlns** module, 228, 273–275
- Modes for filters**, 207–208, 231
- Modules**
  - AAA, 187–188
    - authentication function*, 190–192
    - basic authentication providers*, 188–190, 193–195
    - configuration*, 193
    - digest authentication providers*, 193–195
  - in APR, 54–55
  - in APU, 56–57
  - communicating between, 90–92
  - configuring, 237
    - alternative methods*, 261
    - basics*, 237–239
    - context in*, 255–257
    - custom containers for*, 257–261
    - directives for*, 242–250
    - function types for*, 246–250
    - hierarchy*, 250–255
    - process*, 239–240
    - scope of*, 246
    - server and directory*, 240–241
    - structures for*, 239
    - summary*, 262
    - user data in*, 244–245
  - data management, 88
    - configuration vectors*, 88
    - lifetime scopes*, 88–90
  - debugging. *See* Debugging
  - error handling and reusability of, 172–174
  - libraries for, 118–120
  - for request processing, 168–174
  - written and compiled in other languages, 120–122

**Monitor hook**, 67, 268, 337–338  
**Monitoring connections**, 400  
**Moved Permanently status code**, 415  
**mpm\_common**, hooks exported by, 268  
**Multi-homed Web servers**, 526  
**Multi-Processing Modules (MPMs)**, 22, 25  
   cross-MPM programming issues, 101–102  
     *process and global locks*, 102–104  
     *shared memory*, 104–106  
   for debugging, 331  
   purpose, 26–27  
   UNIX-family, 27–28  
   working with, 28–29  
**Multiline comments**, 87–88  
**multipart/byteranges type**, 519–520  
**multipart/form-data format**, 138  
**Multipart types in HTTP/1.1 specification**, 380–381  
**Multiple Choices status code**, 414–415  
**Multiple responses in cache expiration model**, 437–438  
**MultiViews option**, 159  
**must-revalidate directive**, 468–469  
**Mutexes**, 102–103  
**MySQL databases**, 79  
   drivers, 82–83, 295  
   in LAMP architecture, 289  
   thread-safe library versions, 116  
   DBD Driver, 307–320

## N

**N rule**, 367  
**Names**  
   attacks based on, 507–508  
   HTTP rules, 366  
**Namespace modules**, 273–274  
**Namespacing in APR**, 57–58  
**NDBM**, 80–81  
**Negotiated responses, caching**, 449–450

**Negotiation**  
   content, 158–160, 424–427  
   in persistent connections, 397–398  
**Networks**  
   APR, 76  
   security for, 111  
**New functions, implementing**, 264–266  
**News, websites for**, 19  
**Newsgroups**, 16  
**no-cache directive**, 463  
**No Content status code**, 412–413  
**no-store directive**, 464  
**no-transform directive**, 469  
**Non-authoritative Information status code**, 412  
**Non-shared caches**, 450  
**Nonmodifiable headers**, 446–447  
**Nonstandard request processing**, 44  
**Nonvalidating conditionals**, 444  
**Not Acceptable status code**, 419–420  
**Not Found status code**, 419  
**Not Implemented status code**, 423  
**NOT\_IN\_DIR macro**, 255  
**NOT\_IN\_DIRECTORY macro**, 255  
**NOT\_IN\_FILES macro**, 255  
**NOT\_IN\_LIMIT macro**, 255–256  
**NOT\_IN\_LOCATION macro**, 255  
**NOT\_IN\_VIRTUALHOST macro**, 255  
**Not Modified status code**, 416–417  
**Notational conventions in HTTP/1.1 specification**, 366–369  
**Notes for filters**, 214–215  
**notes table**, 91  
**Null pointers**, 332  
**NULL values**, 269

**O**  
**OK return value**, 126  
**OK status code**, 411  
**Online chat**, 17  
**Online forums**, 16–17

- only-if-cached directive, 468**
- open\_logs hook, 268**
- Operating systems**
  - MPMs with, 28–29
  - for security, 111–113, 178
- Operational phase in two-phase operation, 25**
- Optional elements in HTTP rules, 367**
- optional\_fn\_retrieve hook, 268**
- Optional functions, 265–266**
- Optional hooks, 270–271**
- OPTIONS method, 404–405**
- OR\_ALL option, 246**
- OR\_AUTHCFG option, 246**
- OR\_FILEINFO option, 246**
- OR\_INDEXES option, 246**
- OR\_LIMIT option, 246**
- OR\_OPTIONS option, 246**
- Oracle drivers, 295**
- Order**
  - hook execution, 269
  - request processing, 49–50
- Order directive, 182**
- Origin servers**
  - and 100 (Continue) status, 401
  - in HTTP/1.1 specification, 361
- Output, module, 133–134**
- Output filters, 202**
  - callback functions for, 205, 207
  - strategies for, 210–211
- Overflow, buffer, 109**
  
- P**
- Parameters, media type, 379**
- Parentheses () in rules, 367**
- parse\_form\_from\_POST function, 140–142**
- parseChunk function, 206**
- Parsing**
  - filters for, 221–227
  - form data, 138–144
  - HTTP constructs, 368
  
- Partial Content status code, 413–414**
- Passwords**
  - in Apache 2.1/2.2, 184
  - in basic authentication, 178–179
  - in mod\_authn\_dbd, 305–306
- Patent license, 345**
- Patents, 14–15**
- Paths and path names**
  - attacks based on, 507–508
  - inbound/outbound, 364
  - vs. URLs, 157
- People and processes in security, 178**
- Per-directory configuration, 41**
- Per-server configuration, 41**
- Perchild MPM, 28**
- Perl language, 2, 289**
- Persistent connections**
  - considerations, 399
  - HTTP/1.0 compatibility with, 526–527
  - in message transmissions, 400
  - operation, 397–398
  - proxy servers, 398
  - purpose, 396–397
- Persistent data**
  - garbage collection, 96–98
  - memory/resource management, 96–99
  - reslist, 99
  - resource reuse, 99
  - scope of, 90
  - shared resources, 106
  - subpools, 98
  - thread safety in, 93–96
- Personal information, security for, 505–507**
- Peruser MPM, 28**
- Philosophy of Apache Software Foundation, 6**
- PHP in LAMP architecture, 289**
- Pipe bucket type, 75**
- Pipelining**
  - for filters, 205–206
  - in persistent connections, 398
- Piracy, 14–15**

**Platform and architecture, 21**

- basic concepts and structures, 29
- configuration basics, 41–42
- conn\_rec object, 37–39
- cross-platform issues
  - API builds*, 284–288
  - programming*, 99–101
- for DBD driver, 306
- key API components, 39–41
- MPMs, 26–29
- overview, 21–22
- process\_rec object, 37
- request processing. *See* Request processing and messages
- request\_rec object, 30–35
- server\_rec object, 35–37
- two-phase operation, 22–26

**PMC members in Apache Software Foundation, 5****Pointers**

- crashes from, 332
- declaring, 87
- for merging containers, 253–254
- in request\_rec, 30
- in shared memory, 105–106

**Pool bucket type, 75****poolclass class, 64****Pools, *See* Apache Portable Runtime (APR), pools****post\_config hook, 268****Post-configuration hooks, 66, 268****POST method, 404**

- cache invalidation with, 451
- OK status code with, 411
- parsing data from, 140–142
- vs. PUT, 408
- working with, 407

**post\_read\_request hook**

- purpose, 44
- in request processing, 155–156

**PostgreSQL drivers, 295****Postprocessing filters, 212****Pragma general-header field, 490–491****pre\_config hook, 268****Pre-configuration hooks, 66, 268****pre\_connection hook, 67****pre\_mpm hook, 268****Precautionary principle, 107–109****Precondition Failed status code, 421****Predecessors of functions, 269****Prefork MPM, 27–28****Prematurely closed connections, 402–403****Prepackaged configuration functions, 245–246****Preprocessing filters, 212****Preprocessor directives, 285–286****private directive, 463****Private metadata, 160****Privileges, 111****process\_connection hook, 67****process\_rec object, 29, 37****process\_score entry, 335****Processes, 29**

## APR, 78

- in cross-MPM programming, 102–104

**Processing hooks, 44–46, 50–51****Product tokens, 381****Programming techniques and caveats, 85**

- coding conventions, 85–88
- cross-MPM, 101–106
- cross-platform, 99–101
- external dependencies and libraries, 114–120
- inter-module communication, 90–92
- module data management, 88–90
- modules written and compiled in other languages, 120–122
- persistent data, 93–99
- secure programming, 106–113
- summary, 122
- thread-safe, 92–93

**Protocol filters, 202–205, 215–217****Provider API, 272–277****Providers, authentication, 188–190, 193–195****Proxies**

- and 100 (Continue) status, 402
- Apache, 48
- for cache responses, 446–447

- denial of service attacks on, 510
- in HTTP/1.1 specification, 361
- persistent connections, 398
- security considerations in HTTP, 509–510
- Proxy-Authenticate response-header field, 187, 491**
- Proxy Authentication Required status code, 186, 420**
- Proxy-Authorization request-header field, 491**
- proxy-revalidate directive, 469**
- ProxyHTMLLinks directive, 249**
- ProxyPassReverse directive, 241**
- public directive, 463**
- Public domain software, 12**
- Public mailing lists, 16**
- PUT method**
  - cache invalidation with, 451
  - as idempotent method, 404
  - working with, 408–409
- Python in LAMP architecture, 289**

## Q

- q parameter in Accept field, 454**
- Quality values**
  - in HTTP/1.1 specification, 381–382
  - in RFC 2068, 527
- Queues, 73**
- quick\_handler hook, 44**

## R

- Range field, 492–494**
- Range request-header field, 485, 493–494**
- Range units in HTTP/1.1 specification, 383**
- read function, 74**
- readbytes, 231–232**
- Reading form data, 138–144**
- Reason phrase in response messages, 392–394**
- Redirect directive, 158, 163**

- REDIRECT\_STATUS variable, 162**
- Redirection 3xx status code definitions, 414–418**
- Redirects in request processing, 161–162**
- Redistribution section in Apache license, 345–346**
- Reference manual, APR, 57**
- References in HTTP/1.1 specification, 512–516**
- Referer request-header field, 494**
- register\_hooks function, 117, 189, 192, 267, 269, 271**
- Registering functions, 115**
- Release managers, 9**
- Releases, 8–9**
- Reload controls, 466–469**
- RemoveOutputFilter directive, 211**
- Replacement of caches, 452**
- Report generation in Site Valet, 168**
- Repositories, code, 7, 10**
- Representation in HTTP/1.1 specification, 360**
- Request Entity Too Large status code, 422**
- Request-Line in request messages, 388**
- Request pools, 30**
- Request processing and messages, 151–152, 388**
  - in Apache, 42, 155–156
    - caching in, 160*
    - content generation in, 42–43*
    - content negotiation in, 158–160*
    - data axis and filters, 46–49*
    - hooks in, 156*
    - mapping to filesystem, 156–158*
    - order of, 49–50*
    - phases in, 43–44*
    - pools in, 66*
    - private metadata in, 160*
    - processing hooks, 44–46, 50–51*
    - scope in, 89–90*
    - security in, 160*
    - summary, 51*

**Request processing and messages, *continued***

- in HTTP, 153, 360
  - anatomy of*, 153–155
  - diversion in*, 161–163
  - header fields in*, 391
  - information gathering in*, 163–167
  - logging in*, 161
  - malformed and malicious*, 163
  - Request-Line for*, 388–390
  - resources identified in*, 390–391
- modules for, 168–174
- summary, 174–175

**request\_rec object**

- for configuration data, 239–240
- definition, 30–35
- for filters, 203
- for HelloWorld, 129–130
- for module communication, 90–92

**Request/response protocols, 364****Request Timeout status code, 420****Request-URI**

- with PUT, 408
- in request messages, 389–390

**Request-URI Too Long status code, 422****Request URLs, 43****Requested Range Not Satisfiable status code, 422****Require directive, 186, 188, 256****Requirements in HTTP/1.1 specification, 359****Reset Content status code, 413****Reslists, *see* APR reslists****APR reslists**

- implementing, 278–284
- working with, 99

**Resources and resource management**

- APR, 59
  - lifetime*, 65–67
  - pool limitations*, 68
  - pools*, 61–64, 78–79
  - problem of*, 60–61
- in HTTP/1.1 specification, 360
- in request messages, 390–391
- reusing, 99
- shared, 106

**Response is stale warn code, 503****Response page for HelloWorld, 130–132****Responses and response messages, 153, 360, 392**

- cacheability of, 444–445
- from caches, 445–448
- header fields, 394
- status-line, 392–394

**Responsibility for intellectual property violations, 16****Results sets, 296–297****Retry-After response-header field, 494–495****Return values**

- APR, 58
- modules, 126–127

**Reusability**

- DBD driver, 306
- modules, 172–174
- resources, 99

**Revalidation failed warn code, 503****Revalidation for Cache-Control, 466–469****Review and consensus process, 8****Review-Then-Commit (RTC) code, 8****RFC 2045 entities, 521–524****RFC 2068, changes from, 527–529****Rings, 73****Roles in Apache Software Foundation, 4–5****ROLLBACK for transactions, 297****RSRC\_CONF option, 246****RTC (Review-Then-Commit) code, 8****S****s-maxage directive, 465, 528****Safe methods, 403–404****Safe programming techniques, 178****Satisfy directive, 182****SAX filters, 228****Scope**

- of module configuration, 246
- of module data, 88–90



- Scoreboard, 334–335**
- scoreboard.h file, 334**
- Scrutiny of DBD driver, 306**
- SDBM, 80**
- Seconds, delta, 374**
- Security, 177–178**
  - AAA. *See* AAA (Access, Authentication, and Authorization)
  - HTTP, 504–505
    - authentication credentials and idle clients, 509*
    - content-disposition issues, 509*
    - denial of service attacks on proxies, 510*
    - DNS spoofing, 508*
    - file and path name attacks, 507–508*
    - location header spoofing, 508–509*
    - personal information, 505–507*
    - proxies and caching, 509–510*
  - in request processing, 160
  - secure programming, 106–107
    - denial of service attacks, 109–110*
    - operating system for, 111–113*
    - precautionary principle, 107–109*
- Security phase, hooks in, 156**
- See Other status code, 416**
- Self-configuration filters, 213–215**
- Semantically transparent caches, 363**
- Semi-colons (:), HTTP comments, 368**
- Sensitive information**
  - encoding in URIs, 506
  - transfer of, 505–506
- Separators in coding, 86**
- serf library, 75**
- Server-driven content negotiation, 425–426**
- Server error 5xx status code definitions, 423–424**
- Server log information, abuse of, 505**
- server\_rec object, 29**
  - for configuration data, 239–240
  - definition, 35–37
- Server response-header field, 495**
- Server-specified expiration, 433**
- Servers, 29**
  - configuring, 240–241
  - debugging, 329–331
  - HTTP/1.1, 361
  - multi-homed, 526
  - start-up, 268, 329–331
- Service Unavailable status code, 423–424**
- Services, providing, 277–284**
- Session management with SQL, 196–197**
- set\_allowed\_ports function, 248**
- set\_links function, 249–250**
- set\_links\_raw\_args function, 250**
- setaside function, 74**
- SetHandler directive, 42**
- setjmp function, 114**
- SetOutputFilter directive, 211**
- Shared caches, 450**
- Shared memory, 104–106**
- Shareware, 13**
- Shutdown in two-phase operation, 26**
- Side effects**
  - with GET and HEAD, 451
  - with libraries, 119
- Site Valet software, 53, 168**
- Skeletons, module, 124–126**
- Sloppy programming, 108–109**
- Smart filtering, 211–217**
- Socket bucket type, 75**
- Software licenses, 12–14**
- Specific end-to-end cache revalidation, 467**
- split function, 74**
- Spoofing**
  - DNS, 508
  - location headers, 508–509
- SQL databases**
  - and apr\_dbd, 82–83
  - session management with, 196–197
  - statements in, 295–296
- Square brackets ([]) in rules, 367**
- Stable branches of code repository, 7**
- Stale responses, 363**
- Standard modules, 333–335**

**STANDARD20\_MODULE\_STUFF** macro,  
124, 239–240

**start\_comment** function, 258

**Start-up**, server, 268, 329–331

**Start-up phase**, configuration in, 23–25

**startElement** event, 228

**State changes in libraries**, 117–118

**Static data in thread-safe programming**, 92

**static functions**, 129

**Status code definitions**, 392–394, 410

- client error 4xx, 418–423
- from handlers, 126
- informational 1xx, 410–411
- redirection 3xx, 414–418
- server error 5xx, 423–424
- successful 2xx, 411–414

**STATUS** file, 8

**Status-line in response messages**, 392–394

**stdio-like filter I/O**, 227–230

**Stealth libraries**, 119

**Strings**

- APR, 69
- format, 295–296

**Strong cache validators**, 439–442

**Strong entity tags**, 382

**Structures**. *See* Data structures

**Stubs**, 122

**Submission of contributions section in Apache license**, 346

**Subpools**, 98

**subprocess\_env** table, 91

**Subrequests**, 163–167

**Subversion repository**, 10

**Subversion system**, 7

**Successful 2xx status code definitions**,  
411–414

**Successors of functions in determining order**,  
269

**suexec**, 28

**switch statements**, 87

**Switching Protocols status code**, 411

**System administrators, debugging assistance  
for**, 326–327

**Systems-level modules**, 26

## T

**Tables**, 71–72

- of bags, 138
- hash tables, 72–73

**TCP connections**, 29, 37

**TE request-header field**, 495–496

**Temporary files, creating**, 100–101

**Temporary Redirect status code**, 417–418

**Termination, library**, 116–117

**Terminology in HTTP/1.1 specification**,  
359–365

**test\_config** hook, 268

**Testing, forum participation for**, 11

**Text**

- filtering, 217–221
- media types, 379–380
- parsing, 221–225

**TEXT** rule, 369

**Third-party extensions**, 18

**Third-party intellectual property**, 15–16

**Third-party libraries**

- compatibility of, 114
- debugging, 340–341

**Thread safety**, 92–93

- with libraries, 116
- in persistent data, 93–96

**Threads, APR**, 78

**throw/catch structures**, 116

**Time**

- in HTTP/1.1 specification, 373–374
- module for, 70

**tmpfile** function, 100

**Tokens**

- in HTTP/1.1 specification, 381
- in request messages, 388

- Tolerant applications, 520–521**
  - TRACE method**
    - as idempotent method, 404
    - OK status code with, 411
    - working with, 409
  - Traceback for command handler functions, 330–333**
  - Tracing crashes, 331–332**
  - Trademarks in Apache license, 346**
  - Trailer general field, 497**
  - Trailers in chunked transfer coding, 378**
  - Transactions, 297–298**
  - Transfer codings in HTTP/1.1 specification, 376–378**
  - Transfer-Encoding general-header field, 385–386, 497, 523–524**
  - Transfer-length of messages, 386–387**
  - Transformation applied warn code, 503–504**
  - Transformations**
    - filters for, 212
    - XSLT, 168
  - transforms table, 169**
  - Transient bucket type, 75**
  - translate\_alias\_redir function, 157–158**
  - translate\_name hook**
    - purpose, 44
    - in request processing, 155–156
  - Translation, forum participation for, 11**
  - Transmission requirements for connections, 400–403**
  - Transparent caches, 363**
  - Transparent content negotiation, 427**
  - Transparent proxies, 446–447**
  - Transport-layer security, 178**
  - Trunks in code repository, 7**
  - Trust nothing principle, 107–109**
  - try\_alias\_list function, 158**
  - Tunnels, 362**
  - Tutorials, websites for, 19**
  - Two-phase operation, 22–23**
    - operational phase, 25
    - start-up, 23–25
  - type\_checker hooks, 168**
    - purpose, 45
    - in request processing, 155–156
  - Types**
    - entities, 395–396
    - HTTP messages, 383–384
    - media, 379–381
- ## U
- Unauthorized status code, 186, 418–419**
  - Uniform Resource Identifiers (URIs), 371–372**
    - APR handling, 77
    - encoding sensitive information in, 506
    - in request messages, 389–390
  - UNIX-family MPMs, 27–28**
  - Unlimited shared resources, 106**
  - Unspecified end-to-end cache revalidation, 467**
  - Unsupported Media Type status code, 422**
  - Updates, cache invalidation after, 451–452**
  - Upgrade general-header, 498–499**
  - Upstream/downstream flow, 363**
  - URIs (Uniform Resource Identifiers), 371–372**
    - APR handling, 77
    - encoding sensitive information in, 506
    - in request messages, 389–390
  - URLs**
    - vs. filesystem paths, 157
    - http, 372
    - request, 43
  - US-ASCII character set, 374**
  - Use directive, 259–260**
  - use\_macro function, 260–261**
  - use\_namespace function, 274**
  - Use Proxy status code, 417**
  - Usenet newsgroups, 16**
  - User-Agent request-header field, 499**
  - User-Agent string, 161**
  - User agents**
    - in HTTP/1.1 specification, 361
    - warnings, 431–432

User data in configuration functions,  
244–245

Username in basic authentication, 178–179

Users in Apache Software Foundation, 4–5

Users list, 16

util\_header files, 39

util\_filter.h file, 41, 136, 208–209, 227

util\_ldap\_cache\_init function, 105

util\_ldap.h file, 41

util\_script.h file, 41

Utility factor in handler vs. filter decisions,  
46

## V

### Validation

cache, 438–444

for Cache-Control field, 466–469

input, 107–108

### Validators, 363

### Values

hash table, 72–73

quality, 381–382, 527

return, 58, 126–127

### Variants

of documents, 168–171

in HTTP/1.1 specification, 360–361

### Vary field, 499–500

### Vectors, configuration, 30, 88

### Versions

HTTP, 370–371, 525–529

library, 119–120

Subversion system, 7

### vhost macro, 259

### Via general-header field, 500–501

### Virtual hosts, 238

<VirtualHost> container, 238, 241, 251

## W

### Warning general-header field, 501–504

Warnings for caches, 430–432

Weak cache validators, 439–442

Weak entity tags, 383

Web, login, 180

Web servers, multi-homed, 526

Websites for developers, 17–19

Weights for HTTP quality values, 381

Whitespace in coding, 86

Wildcards in RFC 2068, 528

Worker MPM, 27

worker\_score entry, 335

Wrappers, C, 63

write command, 113

Write-through mandatory in caches, 452

WWW-Authenticate response-header field,  
187, 504

## X

xdefault function, 228

xhtml\_hooks function, 277

XML namespace framework, 272–273

XML\_Parse function, 226

xmlns\_filter function, 225

xmlns.h file, 275

xmlns structure, 275–276

XMLNSUseNamespace directive, 274

XSLT transformation, 168

xstartElement function, 228