	3.3.2	GET56
	3.3.3	HEAD57
	3.3.4	PUT57
	3.3.5	POST58
	3.3.6	TRACE58
	3.3.7	OPTIONS60
	3.3.8	DELETE60
	3.3.9	扩展方法
3.4	状态码	<del>]</del> 62
	3.4.1	100 ~ 199信息性状态码62
	3.4.2	200 ~ 299——成功状态码63
	3.4.3	300 ~ 399重定向状态码
	3.4.4	400~499——客户端错误状态码68
	3.4.5	500 ~ 599——服务器错误状态码69
3.5	首部…	70
	3.5.1	通用首部71
	3.5.2	请求首部72
	3.5.3	响应首部74
	3.5.4	实体首部
3.6	更多信	1息77
第4章	连接	<b>转管理79</b>
第 <b>4 章</b> 4.1		接管理·······79 挂接······80
2075		
2075	TCP É	<u>E接······80</u>
2075	TCP 差	E接·······80 TCP 的可靠数据管道·····80
2075	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4	E接       80         TCP 的可靠数据管道       80         TCP 流是分段的、由 IP 分组传送       81         保持 TCP 连接的正确运行       82         用 TCP 套接字编程       84
2075	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4	E接····································
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4	E接       80         TCP 的可靠数据管道       80         TCP 流是分段的、由 IP 分组传送       81         保持 TCP 连接的正确运行       82         用 TCP 套接字编程       84
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TC	E接       80         TCP 的可靠数据管道       80         TCP 流是分段的、由 IP 分组传送       81         保持 TCP 连接的正确运行       82         用 TCP 套接字编程       84         P性能的考虑       85         HTTP 事务的时延       86         性能聚焦区域       87
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TCI 4.2.1	E接       80         TCP 的可靠数据管道       80         TCP 流是分段的、由 IP 分组传送       81         保持 TCP 连接的正确运行       82         用 TCP 套接字编程       84         P性能的考虑       85         HTTP 事务的时延       86
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TC 4.2.1 4.2.2	E接       80         TCP 的可靠数据管道       80         TCP 流是分段的、由 IP 分组传送       81         保持 TCP 连接的正确运行       82         用 TCP 套接字编程       84         P性能的考虑       85         HTTP 事务的时延       86         性能聚焦区域       87
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TC 4.2.1 4.2.2 4.2.3	E接       80         TCP 的可靠数据管道       80         TCP 流是分段的、由 IP 分组传送       81         保持 TCP 连接的正确运行       82         用 TCP 套接字编程       84         P性能的考虑       85         HTTP 事务的时延       86         性能聚焦区域       87         TCP 连接的握手时延       87
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TC 4.2.1 4.2.2 4.2.3 4.2.4	E接       80         TCP 的可靠数据管道       80         TCP 流是分段的、由 IP 分组传送       81         保持 TCP 连接的正确运行       82         用 TCP 套接字编程       84         P性能的考虑       85         HTTP 事务的时延       86         性能聚焦区域       87         TCP 连接的握手时延       87         延迟确认       88
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TCI 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5 4.2.6 4.2.7	E接80TCP 的可靠数据管道80TCP 流是分段的、由 IP 分组传送81保持 TCP 连接的正确运行82用 TCP 套接字编程84P性能的考虑85HTTP 事务的时延86性能聚焦区域87TCP 连接的握手时延87延迟确认88TCP 慢启动89Nagle 算法与 TCP_NODELAY89TIME_WAIT 累积与端口耗尽90
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TCI 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5 4.2.6 4.2.7	E接80TCP 的可靠数据管道80TCP 流是分段的、由 IP 分组传送81保持 TCP 连接的正确运行82用 TCP 套接字编程84P性能的考虑85HTTP 事务的时延86性能聚焦区域87TCP 连接的握手时延87延迟确认88TCP 慢启动89Nagle 算法与 TCP_NODELAY89TIME_WAIT 累积与端口耗尽90连接的处理91
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TCI 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5 4.2.6 4.2.7	E接80TCP 的可靠数据管道80TCP 流是分段的、由 IP 分组传送81保持 TCP 连接的正确运行82用 TCP 套接字编程84P性能的考虑85HTTP 事务的时延86性能聚焦区域87TCP 连接的握手时延87延迟确认88TCP 慢启动89Nagle 算法与 TCP_NODELAY89TIME_WAIT 累积与端口耗尽90连接的处理91常被误解的 Connection 首部91
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TC 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5 4.2.6 4.2.7 HTTP 4.3.1 4.3.2	E接80TCP 的可靠数据管道80TCP 流是分段的、由 IP 分组传送81保持 TCP 连接的正确运行82用 TCP 套接字编程84P性能的考虑85HTTP 事务的时延86性能聚焦区域87TCP 连接的握手时延87延迟确认88TCP 慢启动89Nagle 算法与 TCP_NODELAY89TIME_WAIT 累积与端口耗尽90连接的处理91常被误解的 Connection 首部91串行事务处理时延92
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TC 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5 4.2.6 4.2.7 HTTP 4.3.1 4.3.2	接80TCP 的可靠数据管道80TCP 流是分段的、由 IP 分组传送81保持 TCP 连接的正确运行82用 TCP 套接字编程84P性能的考虑85HTTP 事务的时延86性能聚焦区域87TCP 连接的握手时延87延迟确认88TCP 慢启动89Nagle 算法与 TCP_NODELAY89TIME_WAIT 累积与端口耗尽90连接的处理91常被误解的 Connection 首部91串行事务处理时延92E接94
4.1	TCP 差 4.1.1 4.1.2 4.1.3 4.1.4 对 TC 4.2.1 4.2.2 4.2.3 4.2.4 4.2.5 4.2.6 4.2.7 HTTP 4.3.1 4.3.2	E接80TCP 的可靠数据管道80TCP 流是分段的、由 IP 分组传送81保持 TCP 连接的正确运行82用 TCP 套接字编程84P性能的考虑85HTTP 事务的时延86性能聚焦区域87TCP 连接的握手时延87延迟确认88TCP 慢启动89Nagle 算法与 TCP_NODELAY89TIME_WAIT 累积与端口耗尽90连接的处理91常被误解的 Connection 首部91串行事务处理时延92