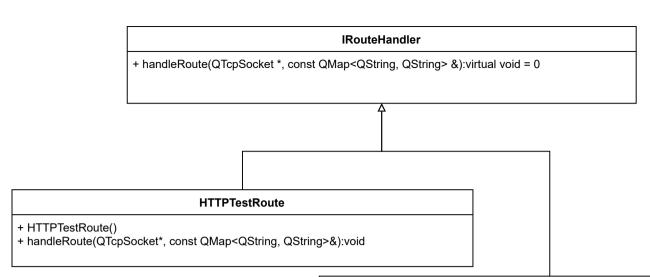


WebSocketClient - *m_webSocket: QWebSocket - *pmm: PlayerComManager - WebSocketClient(QObject*= nullptr) - ~WebSocketClient() + getInstance(): static WebSocketClient& + connectToServer(const QUrl &url): void + sendMessage(const QString &message): void + connected(): void + disconnected(): void + messageReceived(const QString &message): void + error(const QString &message): void - onConnected(): void - onConnected(): void - onConnected(): void - onConnected(): void - onDisconnected(): void - onDisconnected(): void - onDisconnected(): void - onError(QAbstractSocketError error): void



JSONTestRoute

- + JSONTestRoute()
- + handleRoute(QTcpSocket*, const QMap<QString, QString>&):void

WebSocketServer

- m socketServer:QWebSocketServer
- m sockets: static QList<QWebSocket *>
- m instance: static WebSocketServer*
- + WebSocketServer()
- + WebSocketServer(quint16 , QObject*= 0)
- + ~WebSocketServer();
- + broadcast(const QString&):static void
- + getInstance(quint16 = 8080):static WebSocketServer&
- onTextMessageReceived(QString):void
- onNewConnection():void
- onBinaryMessageReceived(QByteArray):void
- onSocketDisconnected():void

JSONUtils

- + JSONUtils()
- + generateJSON(const std::map<std::string, Value>&): static std::string

HttpServer

- m_routes: QMap<QString, Route>
- + HttpServer(QObject *= 0)
- + ~HttpServer()
- + start(quint16): bool
- + errorString(): const QString
- + addRoute(const QString&, const QString&, IRouteHandler *): void

incomingConnection(qintptr): void

WSUtils

- m_allowedMethods: static QStringList
- + WsUtils()
- + createJSONResponse(std::string): static std::string+
- + isAllowedMethod(const QString&): static bool