ASP.NET Web API 2: HTTP MESSAGE LIFECYCLE

HTTP Message Handlers

HTTP message handlers are the first stage in the processing pipeline. They process HTTP request messages on the way in, and HTTP response messages on the way out. To create a custom message handler, derive from the DelegatingHandler class. You can add multiple message handlers. Message handlers can be global or assigned to a specific route. A per-route message handler is invoked only when the request matches that route. Per-route message handlers are configured in the routing table.

Controller

The controller is where you define the main logic for handling an HTTP request. Your controller derives from theApiController class. The controller is where you define the main logic for handling an HTTP request. Your controller derives from theApiController class.

Model Binding

Model binding uses the request to create values for the parameters of the action. These values are passed to the action when the action is invoked.

Controller Action

Invoke Controller Action

Invoke controller action, using HttpActionContext for bindings and model state.

Result Conversion

The return value from the action is converted to an HttpResponseMessage.

HTTP Request

The HTTP request message is first accessed to the HTTP message. The return value from the action is converted to an HttpResponseMessage.

HTTP Response

HTTP message handlers are the first stage in the processing pipeline. They process HTTP request messages on the way in, and HTTP response messages on the way out. To create a custom message handler, derive from the DelegatingHandler class. You can add multiple message handlers. Message handlers can be global or assigned to a specific route. A per-route message handler is invoked only when the request matches that route. Per-route message handlers are configured in the routing table.

Controller

The controller is where you define the main logic for handling an HTTP request. Your controller derives from theApiController class. The controller is where you define the main logic for handling an HTTP request. Your controller derives from theApiController class.

Model Binding

Model binding uses the request to create values for the parameters of the action. These values are passed to the action when the action is invoked.

Controller Action

Invoke Controller Action

Invoke controller action, using HttpActionContext for bindings and model state.

Result Conversion

The return value from the action is converted to an HttpResponseMessage.

HTTP Request

The HTTP request message is first accessed to the HTTP message. The return value from the action is converted to an HttpResponseMessage.

HTTP Response

HTTP message handlers are the first stage in the processing pipeline. They process HTTP request messages on the way in, and HTTP response messages on the way out. To create a custom message handler, derive from the DelegatingHandler class. You can add multiple message handlers. Message handlers can be global or assigned to a specific route. A per-route message handler is invoked only when the request matches that route. Per-route message handlers are configured in the routing table.

Controller

The controller is where you define the main logic for handling an HTTP request. Your controller derives from theApiController class. The controller is where you define the main logic for handling an HTTP request. Your controller derives from theApiController class.

Model Binding

Model binding uses the request to create values for the parameters of the action. These values are passed to the action when the action is invoked.

Controller Action

Invoke Controller Action

Invoke controller action, using HttpActionContext for bindings and model state.

Result Conversion

The return value from the action is converted to an HttpResponseMessage.

HTTP Request

The HTTP request message is first accessed to the HTTP message. The return value from the action is converted to an HttpResponseMessage.

HTTP Response

HTTP message handlers are the first stage in the processing pipeline. They process HTTP request messages on the way in, and HTTP response messages on the way out. To create a custom message handler, derive from the DelegatingHandler class. You can add multiple message handlers. Message handlers can be global or assigned to a specific route. A per-route message handler is invoked only when the request matches that route. Per-route message handlers are configured in the routing table.

Controller

The controller is where you define the main logic for handling an HTTP request. Your controller derives from theApiController class. The controller is where you define the main logic for handling an HTTP request. Your controller derives from theApiController class.

Model Binding

Model binding uses the request to create values for the parameters of the action. These values are passed to the action when the action is invoked.

Controller Action

Invoke Controller Action

Invoke controller action, using HttpActionContext for bindings and model state.

Result Conversion

The return value from the action is converted to an HttpResponseMessage.

HTTP Request

The HTTP request message is first accessed to the HTTP message. The return value from the action is converted to an HttpResponseMessage.

HTTP Response

HTTP message handlers are the first stage in the processing pipeline. They process HTTP request messages on the way in, and HTTP response messages on the way out. To create a custom message handler, derive from the DelegatingHandler class. You can add multiple message handlers. Message handlers can be global or assigned to a specific route. A per-route message handler is invoked only when the request matches that route. Per-route message handlers are configured in the routing table.

Controller

The controller is where you define the main logic for handling an HTTP request. Your controller derives from theApiController class. The controller is where you define the main logic for handling an HTTP request. Your controller derives from theApiController class.

Model Binding

Model binding uses the request to create values for the parameters of the action. These values are passed to the action when the action is invoked.

Controller Action

Invoke Controller Action

Invoke controller action, using HttpActionContext for bindings and model state.

Result Conversion

The return value from the action is converted to an HttpResponseMessage.