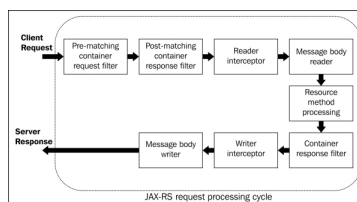




Understanding the JAX-RS resource lifecycle

Before winding up this chapter, let's take a quick look at the lifecycle of the JAX-RS components on the server when a client makes a RESTful web API call. This discussion would be a good summary of the topics that we learned so far on JAX-RS.

The following diagram depicts the sequence of actions taking place on the server when a client invokes the JAX-RS RESTful web service:



Here are the steps:

1. For an incoming REST API call, the container identifies the Java servlet configured for handling the REST API calls by parsing the URI and then delegates the request to the designated servlet.
2. The servlet initializes the JAX-RS runtime and kicks off the RESTful web service request processing cycle for the REST API call.
3. The JAX-RS runtime processes the filters and interceptors configured in the application in the following sequence:
 1. Runtime executes prematching filters (`ContainerRequestFilter` with the `@PreMatching` annotation), which happens before resolving the resource method. The prematching filters are useful if you want to influence resource method resolution.
 2. The next step is to identify the resource method for serving the request. After the resolution of the resource method, postmatching filters are executed (filters without `@PreMatching`). Postmatching filters are useful for performing an authentication check or for performing a basic sanity check on the request parameters.
 3. After executing all filters, the JAX-RS runtime invokes `ReaderInterceptor`. The reader interceptors are useful for manipulating the inbound request stream. A common use case may be to unzip the compressed stream before converting it into Java objects.
 4. The next phase in the cycle is to convert the inbound stream into the Java object representation. This is done by the matching the `MessageBodyReader` provider identified by the runtime.
4. After taking the incoming request through the configured filters, interceptors, and message body readers, the JAX-RS runtime invokes the resource class method identified for serving the request. If you have any Bean Validation annotation configured on the method parameters, it will get executed at this stage:
 1. If the validation fails and results in `ConstraintViolationException`, the runtime skips the execution of the method and identifies `ExceptionHandler` to handle the exception. The exception mapper generates the HTTP response content for the exception. The control flow proceeds to step 5.
 2. Upon successful validation, the resource method is executed.
 3. If the resource method throws some exception, the framework invokes the matching exception mapper to generate the HTTP response body for the exception and skips to step 5.
 4. Upon the successful execution of the method, the framework reads the response object returned by the method.
5. The runtime takes the response content through the following stages before sending it to the client:
 1. The runtime executes `ContainerResponseFilter`, which can be used for adding the desired response header such as cache parameters and content type.



- 2. The runtime executes `WriterInterceptors` for the response content. The interceptor class can hold logic for manipulating the response before sending it to the client, such as zipping the response body.
- 3. As the next step, the runtime invokes the appropriate `MessageBodyWriter`, which writes the Java type to an HTTP message to send it over HTTP. `MessageBodyWriter` can optionally amend or add the HTTP response headers. For instance, `200 OK` is added to the response header if the method execution went well and returned an object. If the method is void, `204 No Content` is set.
- 6. Once the response is ready, the container passes the response back to the client.

NOTE

Note that the default scope of the root resource classes is a request. This means that a new instance of a root resource class is created for serving the next request.

[Recommended](#) / [Queue](#) / [History](#) / [Topics](#) / [Tutorials](#) / [Settings](#) / [Get the App](#) / [Sign Out](#)

 [PREV](#)
[Understanding filters and interceptors in JAX-RS](#)

[NEXT](#) 
[Summary](#)

