

Jakarta EE: The First Parts

Introduction to Prime Jakarta EE APIs

Date: 7th November 2020

Time:

Duration: 40 minutes

Hashtag on Twitter: #jjug_ccc



Today's Speaker



HASUNUMA Kenji

Service Engineer

@khasunuma

Table of Contents

1. Where's main method of Jakarta EE application?
2. Jakarta REST
3. Jakarta CDI
4. MicroProfile Config (See Example)

Examples: <https://github.com/khasunuma/jee-first-parts>

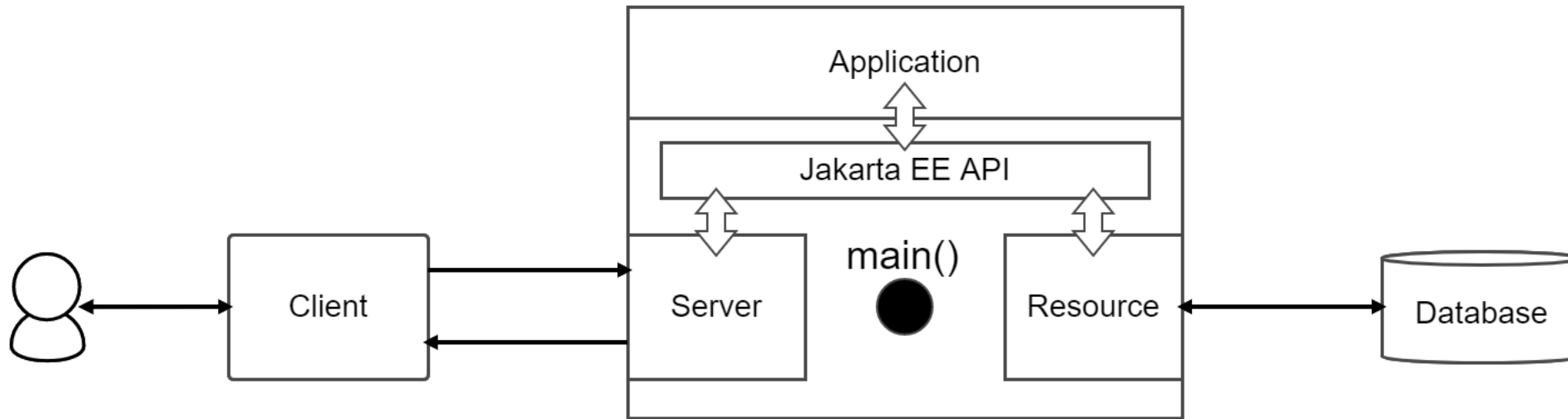


Introduction

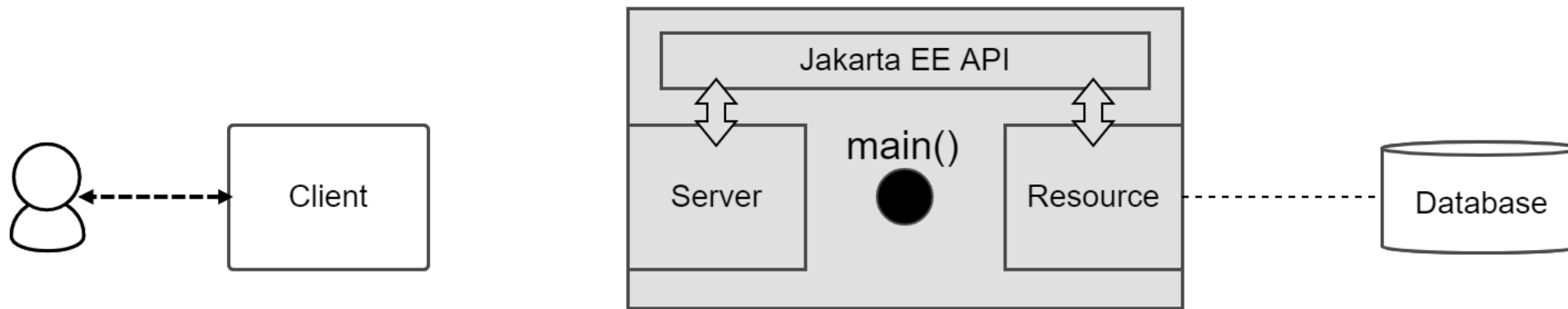
Where's main method of Jakarta EE application?



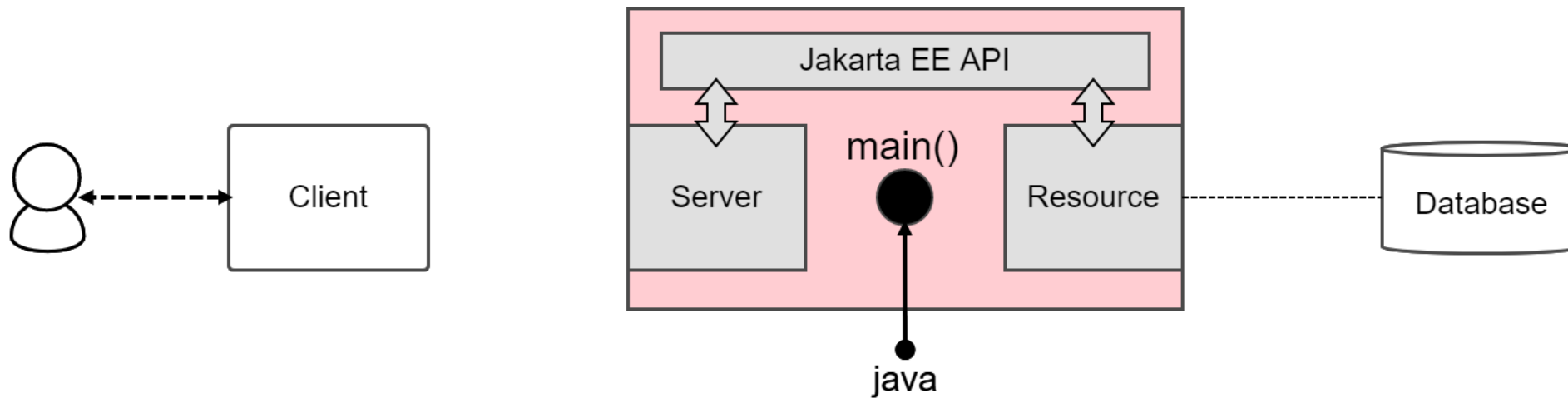
Jakarta EE Architecture



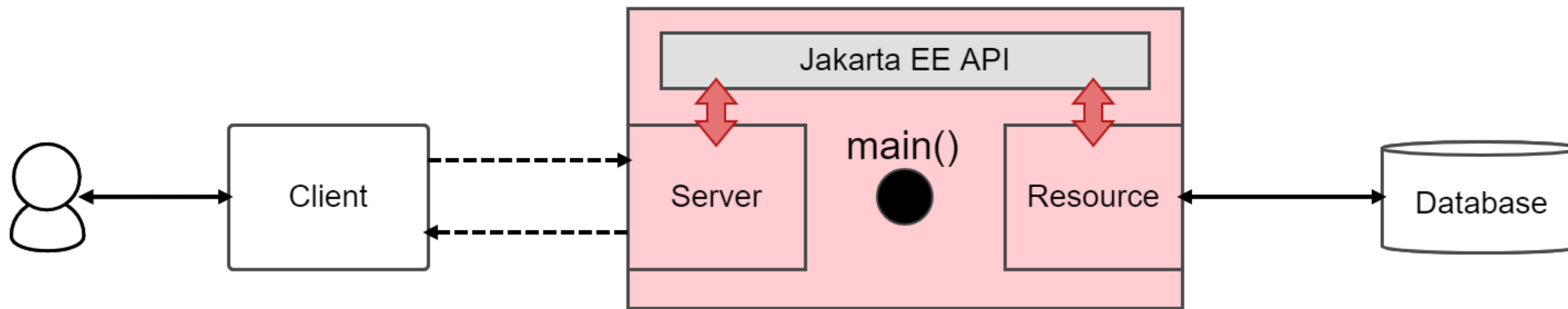
Jakarta EE Architecture



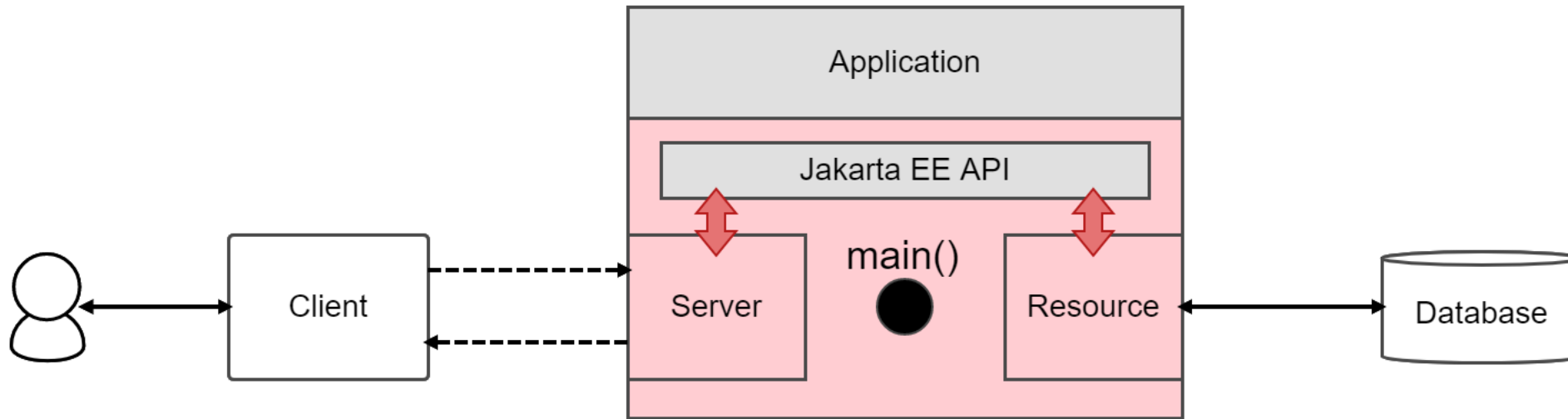
Jakarta EE Architecture



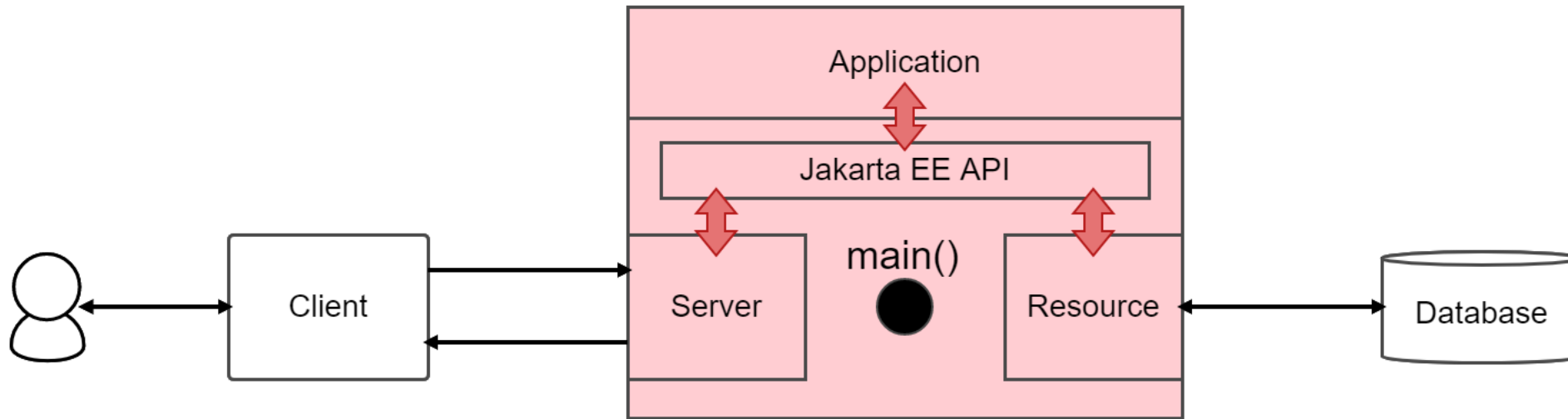
Jakarta EE Architecture



Jakarta EE Architecture



Jakarta EE Architecture



Dependency of Jakarta EE Platform

```
<dependency>  
  <groupId>jakarta.platform</groupId>  
  <artifactId>jakarta.jakartaee-api</artifactId>  
  <version>8.0.0</version>  
  <scope>provided</scope>  
</dependency>
```



Jakarta REST

Where we and Jakarta EE meet



HASUNUMA Kenji @khasunuma



What's Jakarta REST

Jakarta REST (Jakarta RESTful Web Services) is a technology to handle each HTTP requests:

- Easy to use
- Intuitive, Flexible and Expandable
- In most cases, better alternative of Servlets



What's REST

- Primitive idea for Web Applications
 - See in detail: Fielding, T., “Architectural Styles and Design of Network-based Software Architectures”, 2000

The 3 elements of REST:

- Resource – What (HTML, XML, JSON, Image, etc.)
- URL – Where
- Method – How (GET, POST, PUT, DELETE, etc.)



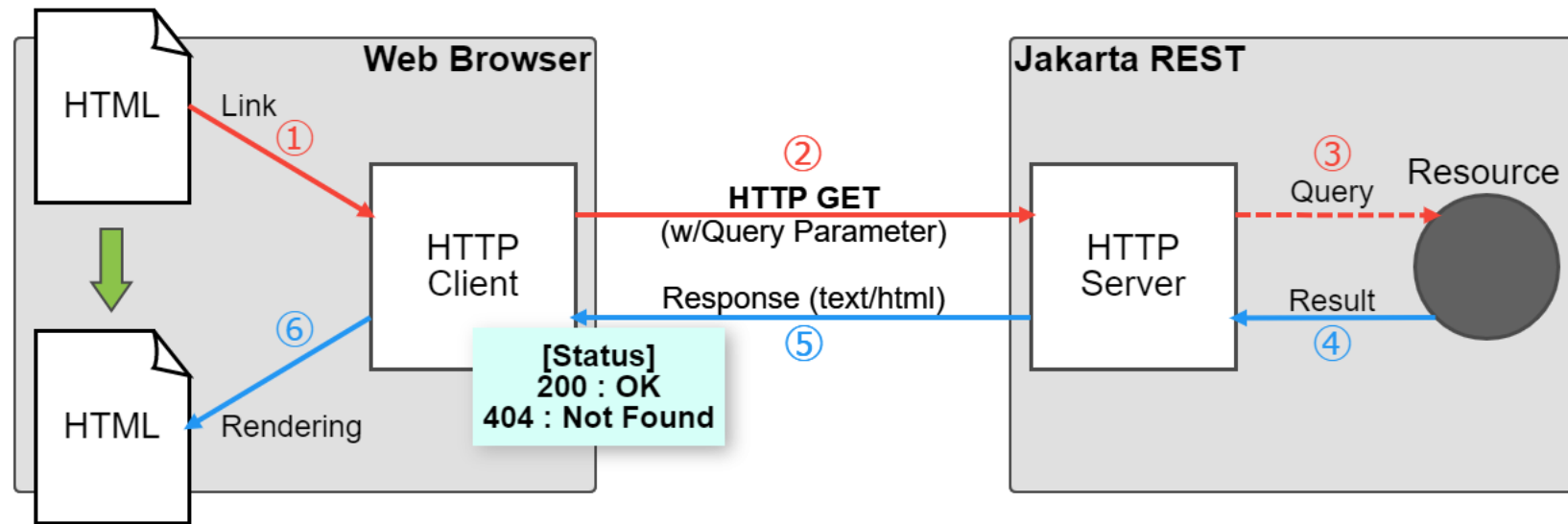
URL syntax – in case of Jakarta EE

http://hostname:8080/context-root/path?query

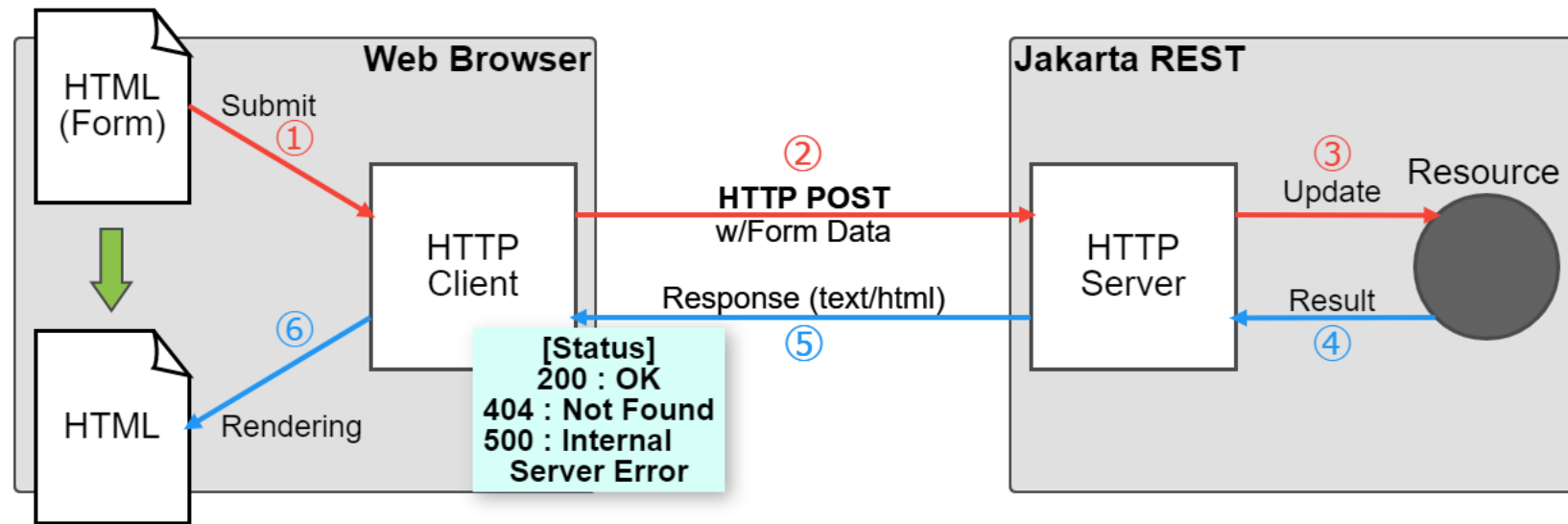
Protocol “http” / “https”	Port Number (Omissible)	The path element to identify which application	Parameters (Omissible)
------------------------------	----------------------------	--	---------------------------



How to send and receive HTTP GET



How to send and receive HTTP POST



Register Jakarta REST

```
@ApplicationPath("app")  
public class MyApplication extends Application {  
  
}
```



Resource Class

```
@Path("some") << URL: "/context-root/app/some" >>
public class SomeResource {
    @GET << Method >>
    << Request >> @Consumes("application/x-www-form-urlencoded")
    << Response >> @Produces("text/html") << Mapping >>
    public String handle(@QueryParam("id") String id) {
        ... // Create a response
    }
}
```

Diagram annotations:

- A red arrow points from `<< Request >>` to `@Consumes("application/x-www-form-urlencoded")`.
- A blue arrow points from `<< Response >>` to `@Produces("text/html")`.
- A red arrow points from `<< Mapping >>` to `String id` in the method signature.



Handle GET method

```
@GET
```

```
@Consumes("application/x-www-form-urlencoded")
```

```
@Produces("text/html")
```

```
public String handle(@QueryParam("id") String id) {  
    // 1. Query Parameter ("id") -> Argument ("id") [0..N]  
    // 2. Create a response  
    // 3. Return value (String) -> Response (HTML)  
    ...  
}
```



Handle POST method

@POST

@Consumes("application/x-www-form-urlencoded")

@Produces("text/html")

```
public String handle(@FormParam("data") String data) {  
    // 1. Form Parameter ("data") -> Argument ("data") [1..N]  
    // 2. Create a response  
    // 3. Return value (String) -> Response (HTML)  
    ...  
}
```



Basic Error Handling

Throw the following exception if an error is occurred:

- `WebApplicationException` – specified HTTP Status
 - `NotFoundException` (HTTP 404)
 - `InternalServerErrorException` (HTTP 500)
 - etc.

No exception, then respond successfully (HTTP 200)



Other Jakarta REST Features

- XML / JSON Binding Integration – for Web Services
- HTTP Client API
- Server Sent Events
- Use ServletContext – for low-level access
- Extensions of each implementation (Not standard)



Jakarta CDI

How to divide a complex thing to simple parts



What's Jakarta CDI

Jakarta CDI is a technology that couples each components:

- View / User Interface
- Compute / Business Logic
- External Interface (e.g. Database, Messaging)
- Structure Value (e.g. Session Object)

etc.



Benefits of Jakarta CDI

- Split components as suitable size
- Choose “**Alternative**” (additional configuration is required)
 - e.g. stub for test environment
- Combine components with different lifecycles (“**Scope**”)
 - **@RequestScoped** – created for each request
 - **@SessionScoped** – created and kept between a HTTP session
 - **@ApplicationScoped** – created and kept until end of the application
- Call implicitly pre-processes and post-processes (“**Interceptor**”)
 - e.g. **@Transactional** – begin and commit/rollback transaction



Dependency Injection by CDI

```
@Named // Qualifier  
@RequestScoped // Scope  
public class CdiBean1 {  
    @Inject // Injection Point  
    private CdiBean2 bean2;  
  
    public void compute() {  
        bean2.execute();  
    }  
}
```

CDI Bean

Inject an instance

```
@Named // Qualifier  
@RequestScoped // Scope  
public class CdiBean2 {  
  
    public void execute() {  
        ...  
    }  
}
```

CDI Bean



Integration REST with CDI

```
@Path("some") Resource / CDI Bean
@RequestScoped // Scope
public class SomeResource {
    @Inject // Injection Point
    private Bean bean;
    @get
    @Produces("text/html")
    public String handle() {
        return bean.execute();
    }
}
```

Inject an instance

```
@Named // Qualifier
@RequestScoped // Scope
public class Bean {
    public String execute() {
        ...
    }
}
```



Next Steps

- Jakarta EE provides various APIs for your applications:
 - Jakarta Persistence – Database Access
 - Jakarta Messaging – Message Queue Access
 - Jakarta Mail – Connection to Mail Servers
 - Jakarta Server Faces – Framework for creating Rich UI
 - Jakarta Security – Security Features
- How to learn Jakarta EE APIs?
 - Each application only uses some part of Jakarta EE
 - You may learn when it is needed for you



Payara Reef: Community Growth Program

Let us help you spread the word about our open source software. Join the Reef!

We'll Support You With:

- Event, JUG, conference sponsorship
- Freebies, swag, handouts, speakers
- Promotion and advertising of events and articles
- Community forum

Learn More:

www.payara.fish/reef

Thank You

Download the open source software:
<https://payara.fish/downloads>

Need support for the Payara Platform?
<https://payara.fish/support>