



Introduction



Part 1 - The Basics



Part 2 - Core MVC 1.0 Features



Part 3 - Useful MVC 1.0 Features



Summary



Part 3

Useful MVC 1.0 Features



Security

Cross Site Request Forgery



```
@Controller
@Path("csrf")
public class HelloController {

    @CsrfValid
    @POST
    public Response post(@FormParam("greeting") String greet {

    }
}
```

Cross Site Request Forgery



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>MVC 1.0 Hello Demo</title>
  </head>
  <body>
    <form action="csrf" method="post" accept-charset="utf-8">
      <input type="submit" value="Click here"/>
      <input type="hidden" name="{mvc.csrf.name}"
        value="{mvc.csrf.token}"/>
    </form>
  </body>
</html>
```



Scopes

@RedirectScoped



```
@Named("greeting")
@RedirectScoped
public class Greeting {

    private String message;

    public void setMessage(String message) {
        this.message = message;
    }

    public void getMessage() { return message; }
}
```

@RedirectScoped



@RequestScoped

```
@Controller  
public class HelloController {
```

```
    @Inject  
    private Greeting greeting;
```

@RedirectScoped

```
    @POST  
    @Path("from")  
    public String hello() {  
        greeting.setMessage("Hello Atlanta!");  
        return "redirect:/to";  
    }  
}
```


@RedirectScoped



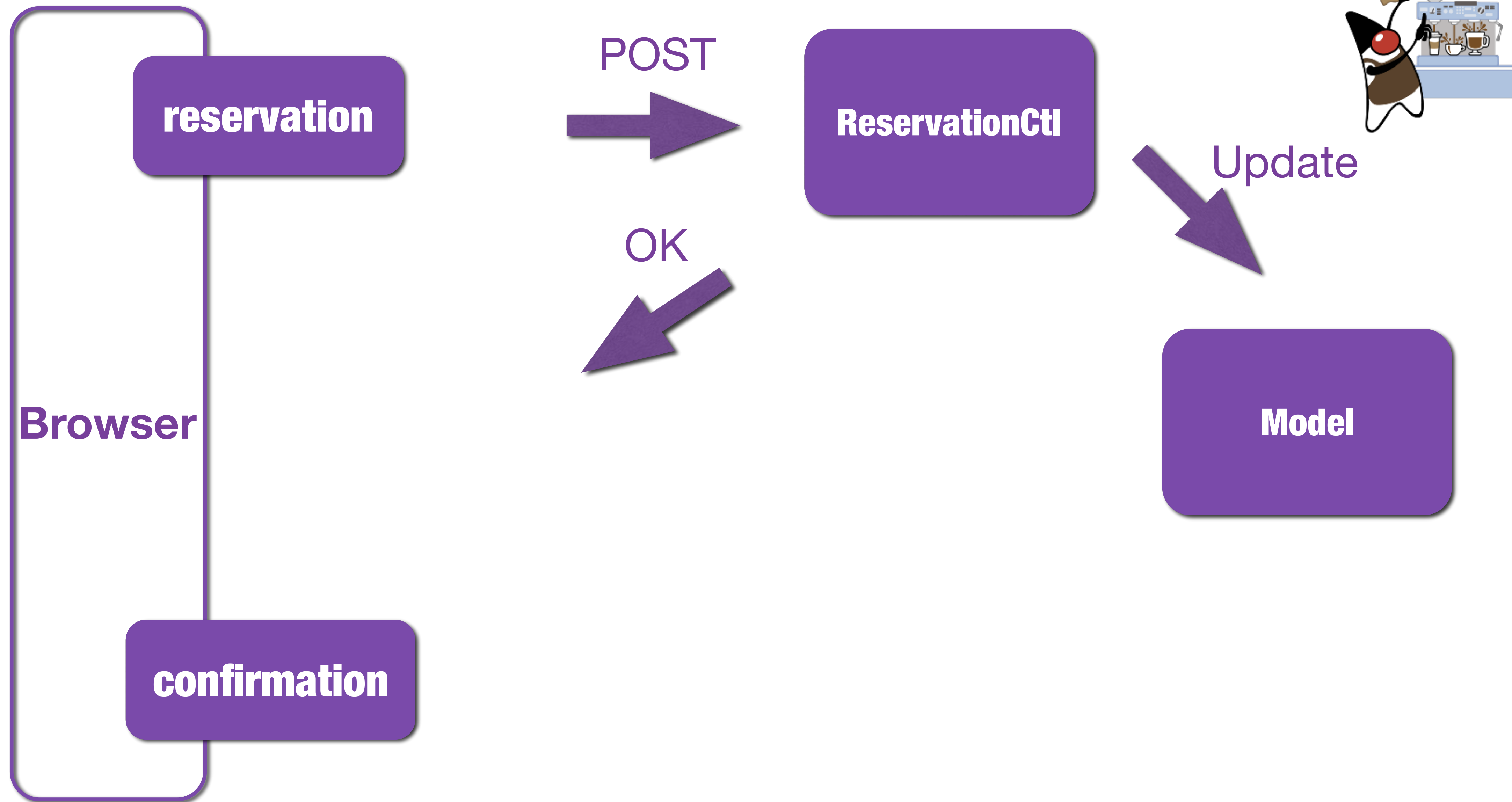
```
@Controller
public class HelloController {

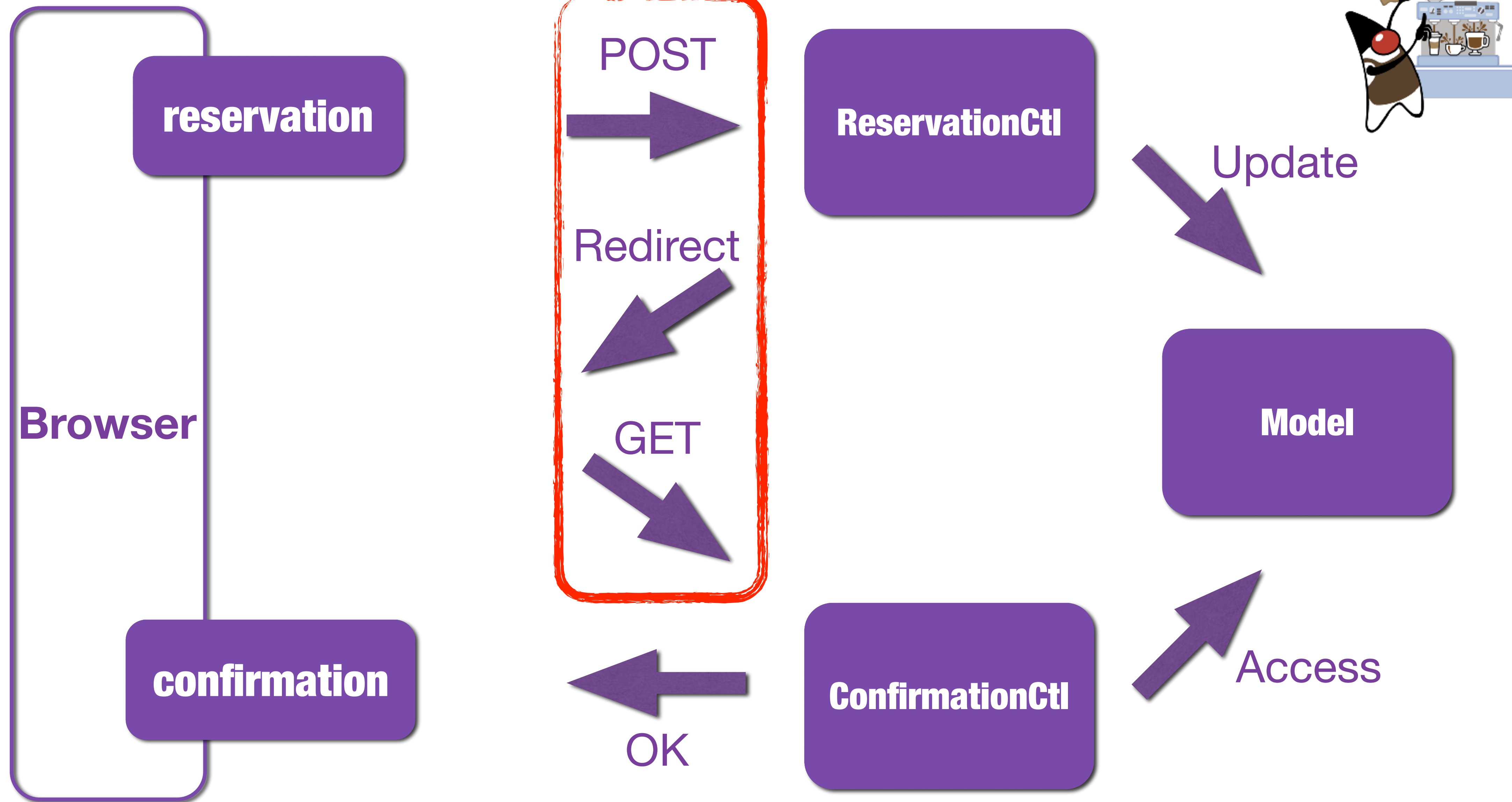
    @Inject
    private Greeting greeting;

    @POST
    @Path("to")
    public String hello() {

        return "here.jsp";
    }
}
```

Same Bean







Internationalization

Controller



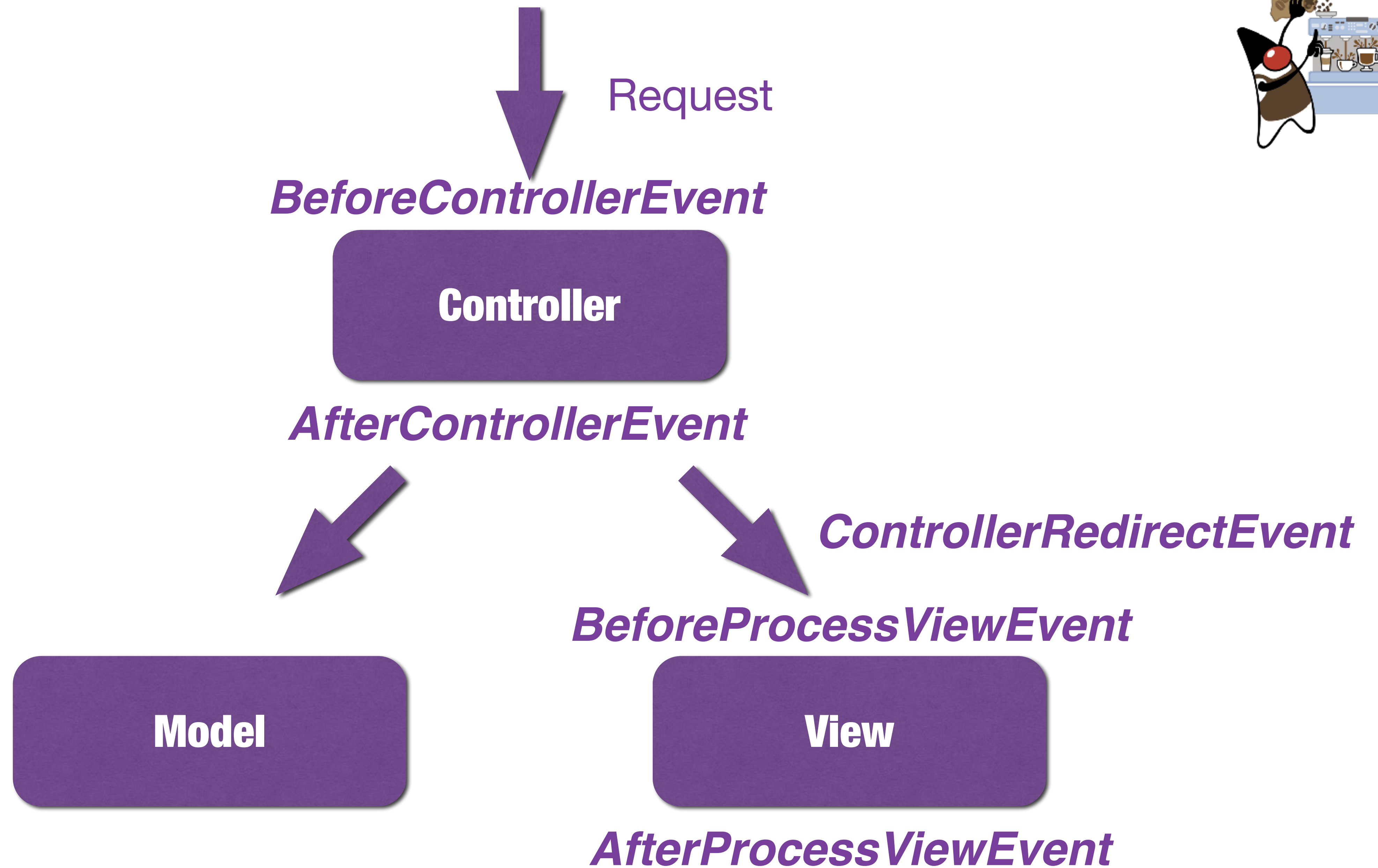
```
@Controller
@Path("hello")
public class HelloController {

    @Inject
    private MvcContext mvc;

    @GET
    public String get() {
        Locale locale = mvc.getLocale();
    }
}
```



Events



Events Example



```
@ApplicationScoped
public class EventObserver {

    void onBeforeController(@Observes BeforeControllerEvent e) {
        println("URI: " + e.getUriInfo().getRequestURI());
    }

    void onAfterController(@Observes AfterControllerEvent e) {
        println("Controller: " +
                e.getResourceInfo().getResourceMethod());
    }
}
```




View Engines



Bring Your Own View Engine

ViewEngine



```
public interface ViewEngine {  
  
    boolean supports(String view);  
  
    void processView(ViewEngineContext context)  
        throws ViewEngineException;  
}
```

View Engine Conflicts



```
@Priority(OUTRAGEOSLY_HIGH)
@ApplicationScoped
public class FreemarkerViewEngine extends ViewEngineBase {
    ...
}
```



Part 1

<https://github.com/ivargrimstad/mvc-hol>

cybercom.com