

Inside the DispatcherServlet.service(...) :

```

Controller controller = applicationContext.getBean("viewOurMissionController", Controller.class);
ModelAndView mav = controller.handleRequest(req, resp);
  
```

```

abstract class AbstractAnnotationConfigDispatcherServletInitializer
implements WebApplicationInitializer {
void onStartup(ServletContext ctx) {
// implementation logic
}
abstract Class<?>[] getRootConfigClasses();
abstract Class<?>[] getServletConfigClasses();
abstract String[] getServletMappings();
}
  
```

```

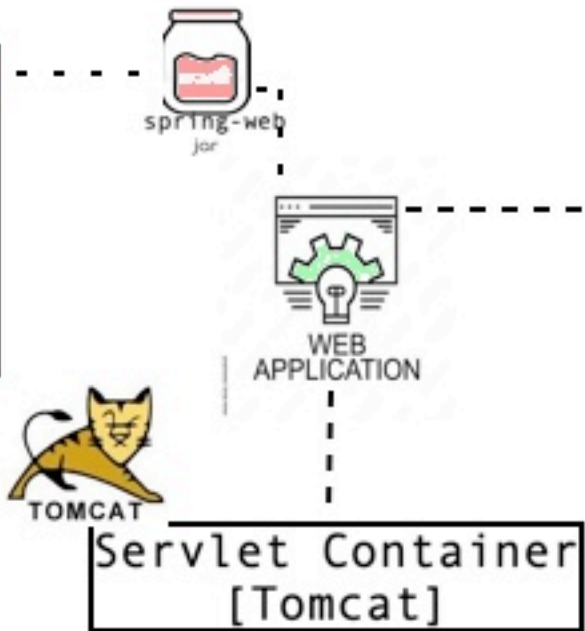
interface WebApplicationInitializer {
void onStartup(ServletContext ctx);
}
  
```

```

class AW2AnnotationConfigDispatcherServletInitializer extends
AbstractAnnotationConfigDispatcherServletInitializer {
Class<?>[] getRootConfigClasses() {...};
Class<?>[] getServletConfigClasses() {...};
String[] getServletMappings() {return new String[]{"*.htm"};}}
  
```

```

@Handlers(WebApplicationInitializer.class)
class SpringServletContainerInitializer
extends ServletContainerInitializer{
void onStartup(Set<Class<?>> handlers,
ServletContext context){...}
}
  
```



1. clicking on hyperlink (GET)
 2. form-submissions (POST)
- #2 allowed http methods: GET, POST
PATCH, PUT, DELETE, HEAD, OPTIONS, TRACE, CONNECT

during the deployment will invoke the onStartup by passing the reference of

- handlers (WebApplicationInitializers)
- ServletContext

```

interface Controller {
ModelAndView handleRequest(req, resp);
}
  
```

There is a benefit of using AbstractController rather than using Controller interface

```

abstract class AbstractController implements Controller {
public ModelAndView handleRequest(HttpServletRequest req, HttpServletResponse resp) {
#1. check for http requestMethod, and allow only GET OR POST
String reqMethod = req.getMethod();
if(reqMethod.eq("GET") || reqMethod.eq("POST")) {
return handleRequestInternal(req, resp);
}else {
// http error
}
#2. is the user HttpSession is valid or not (if requiredSession is enabled)
}
abstract ModelAndView handleRequestInternal(HttpServletRequest req, HttpServletResponse resp);
}
  
```

```

class ViewOurMissionController extends AbstractController {
ModelAndView handleRequestInternal(HttpServletRequest req, HttpServletResponse resp) {
// logic to process the request
return new ModelAndView("our-mission");
}
}
  
```