

Spring Boot Defaults

Data Source Config

DataSourceConfig.java:

```
@Value("${driverClassName}")
private String driverClassName;
@Value("${url}")
private String url;
@Value("${username}")
private String username;
@Value("${password}")
private String password;

@Bean
public static PropertySourcesPlaceholderConfigurer
propertySourcesPlaceholderConfigurer() {
    ...
}

@Bean
public DataSource dataSource() {
    ...
    mysqlDS.setDriverClassName(driverClassName);
    mysqlDS.setUsername(username);
    mysqlDS.setPassword(password);
    mysqlDS.setUrl(url);
    return mysqlDS;
}
```

application.properties:

```
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.url=jdbc:mysql://localhost:3306/
saga?useSSL=false
```

Data Source Config

DataSourceConfig.java:

```
@Bean
public PlatformTransactionManager
transactionManager(EntityManagerFactory
entityManagerFactory) {
    ...
}

@Bean
public Properties hibernateProperties() {
    hibernateProp.put("hibernate.dialect",
"org.hibernate.dialect.MySQL5InnoDBDialect");
    hibernateProp.put("hibernate.hbm2ddl.auto",
"update");
    hibernateProp.put("hibernate.format_sql",
false);
    ...
    return hibernateProp;
}

@Bean
public EntityManagerFactory {
    ...
    return
factoryBean.getNativeEntityManagerFactory();
}
```

application.properties:

```
spring.datasource.platform=org.hibernate.dialect.M
ySQL5InnoDBDialect
```

Application Config

GameConfig.java:

```
@Configuration
@ComponentScan(basePackages = {"rs.saga.service"})
@Import(DataSourceConfig.class)
@EnableJpaRepositories(basePackages =
{"rs.saga.dao"})
public class GameConfig {

    @Bean
    public IGameService gameService(ITeamRepo
teamRepo) {
        GameService gameService = new
GameService();
        gameService.setRepo(teamRepo);
        return gameService;
    }
}
```

@SpringBootApplication:

```
@SpringBootApplication
public class SagaappApplication {
    public static void main(String[] args) {
        ...
    }
}
```

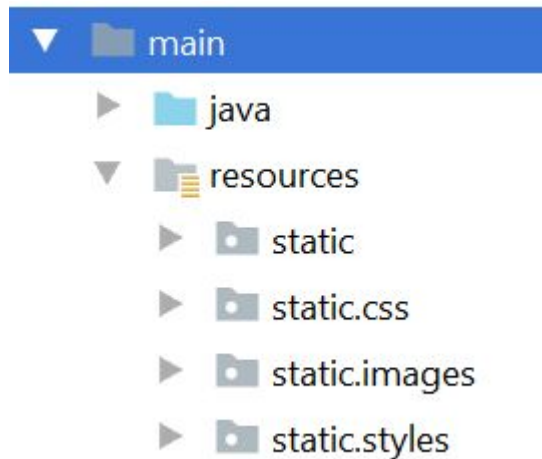
Web Config

WebConfig.java:

```
public class WebConfig implements WebMvcConfigurer
{
    //Declare our static resources
    @Override
    public void
addResourceHandlers(ResourceHandlerRegistry
registry) {

        registry.addResourceHandler("/images/**")
.addResourceLocations("/images/");

        registry.addResourceHandler("/styles/**")
.addResourceLocations("/styles/");
    }
}
```

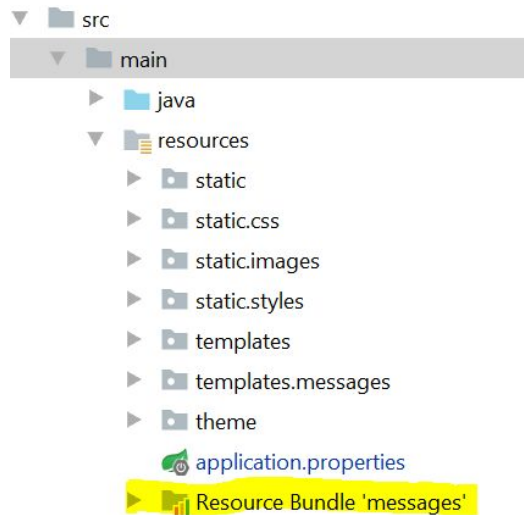


Web Config

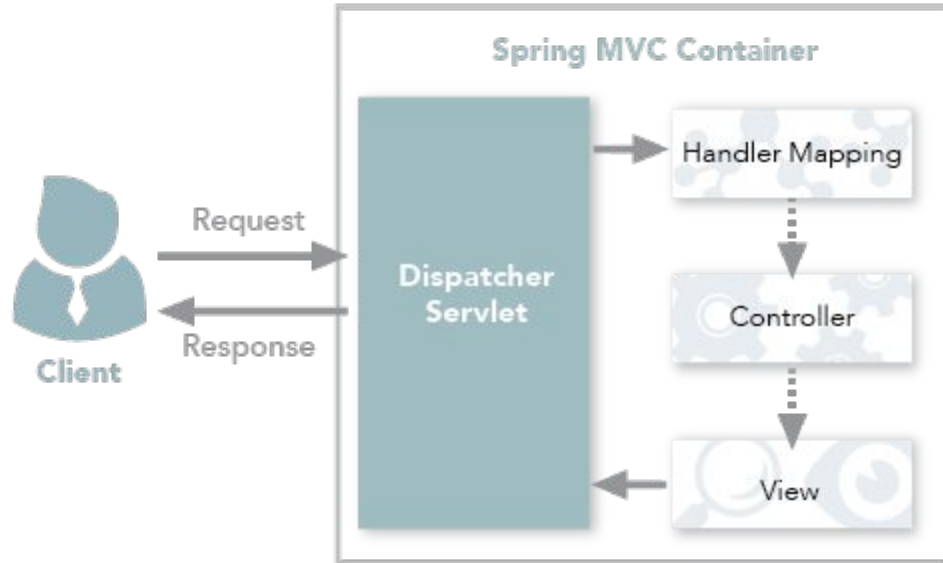
WebConfig.java:

```
@Bean
MessageSource messageSource() {
    ReloadableResourceBundleMessageSource messageSource
    = new ReloadableResourceBundleMessageSource();

    messageSource.setBasename("/WEB-INF/messages/global");
    messageSource.setDefaultEncoding("UTF-8");
    messageSource.setCacheSeconds(1);
    return messageSource;
}
```



View Resolvers



Multiple View Resolvers

```
@Bean
public ViewResolver resourceBundleViewResolver() {

    ResourceBundleViewResolver viewResolver = new
    ResourceBundleViewResolver();
    viewResolver.setBasename("views");
    viewResolver.setOrder(0);
    return viewResolver;
}
```

```
home.(class)=org.springframework.web.servlet.view.
JstlView
auth.(class)=org.springframework.web.servlet.view.
JstlView
players/show.(class)=org.springframework.web.servl
et.view.JstlView
players/list.(class)=org.springframework.web.servl
et.view.JstlView
players/form.(class)=org.springframework.web.servl
et.view.JstlView
auth.url=/WEB-INF/auth.jsp
home.url=/WEB-INF/home.jsp
players/show.url=/WEB-INF/players/show.jsp
players/list.url=/WEB-INF/players/list.jsp
players/form.url=/WEB-INF/players/form.jsp
```