

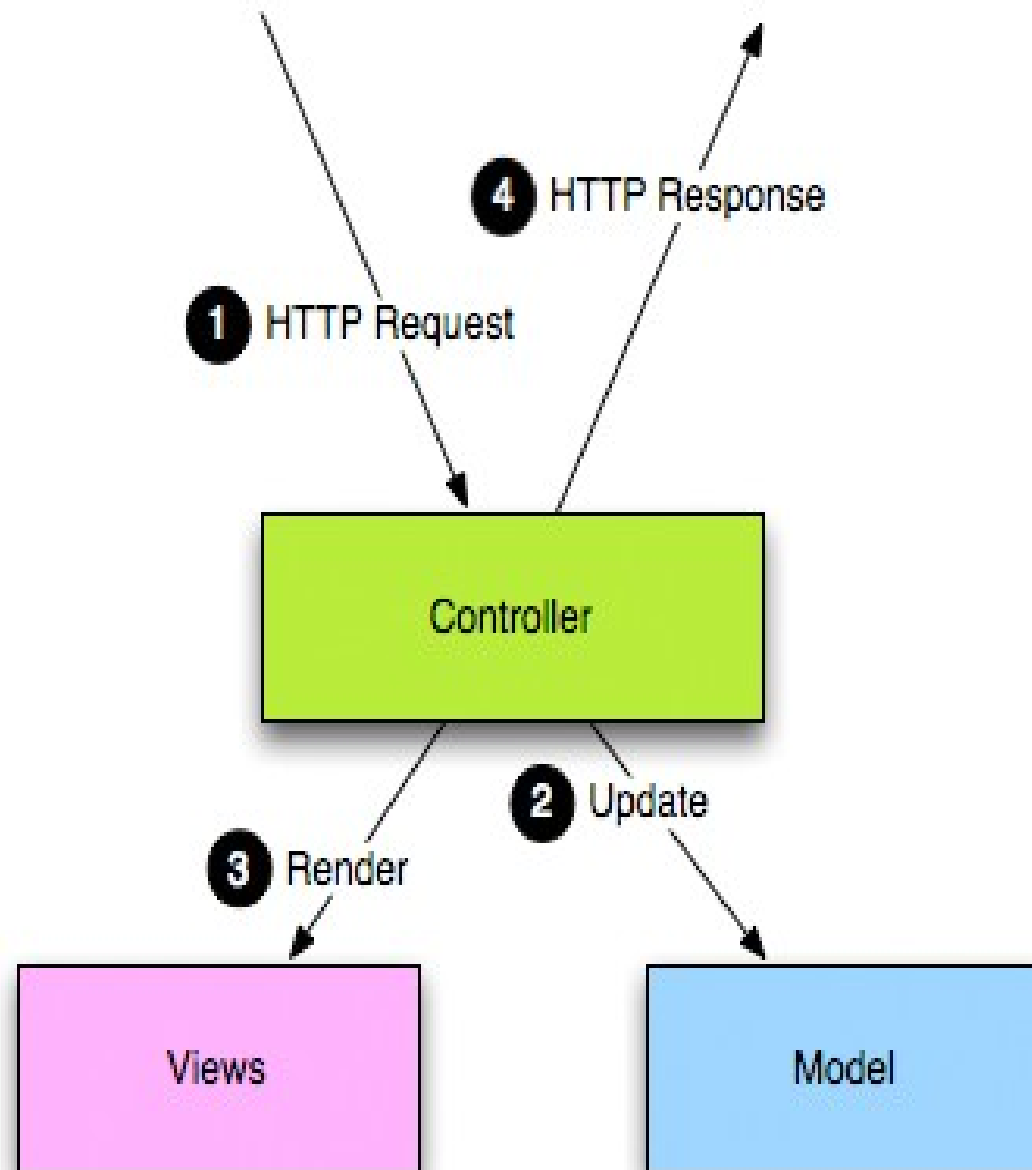
Play!



The Play framework

- Wprowadzenie
- Co w nim polubimy
- Dla kogo on jest
- Czego nie polubimy
- Moduły
- Testowanie
- Widok
- Struktura aplikacji





```
response.cacheFor("10mn");
```



Bindowanie parametrów HTTP do zmiennych



Bindowanie parametrów HTTP do zmiennych

/articles/archive?date=08/01/08&page=2

```
public static void archive(Date date, Integer page) {  
    List<Article> articles = Articles.fromArchive(date, page);  
    render(articles);  
}
```



Bindowanie parametrów HTTP do zmiennych

/articles/17/details

/articles/{articleId}/details

ArticleController.displayDetails

```
public static void displayDetails(Integer articleId) {  
    List<Article> articles = Articles.fromArchive(page);  
    render(articles);  
}
```



DRY przy przekazywaniu parametrów do widoku
- bardziej sucho się nie da

```
Article article = Article.findById(id);  
User user = User.getConnected();  
Map<String, Object> model = new HashMap<String, Object>();  
model.put("article", article);  
model.put("user", user);  
render(model);
```



DRY przy przekazywaniu parametrów do widoku
- bardziej sucho się nie da

```
Article article = Article.findById(id);  
User user = User.getConnection();  
Map<String, Object> model = new HashMap<String, Object>();  
model.put("article", article);  
model.put("user", user);  
render(model);
```



DRY przy przekazywaniu parametrów do widoku
- bardziej sucho się nie da

```
Article article = Article.findById(id);  
User user = User.getConnection();  
render(article, user);
```



JPA gotowe do akcji

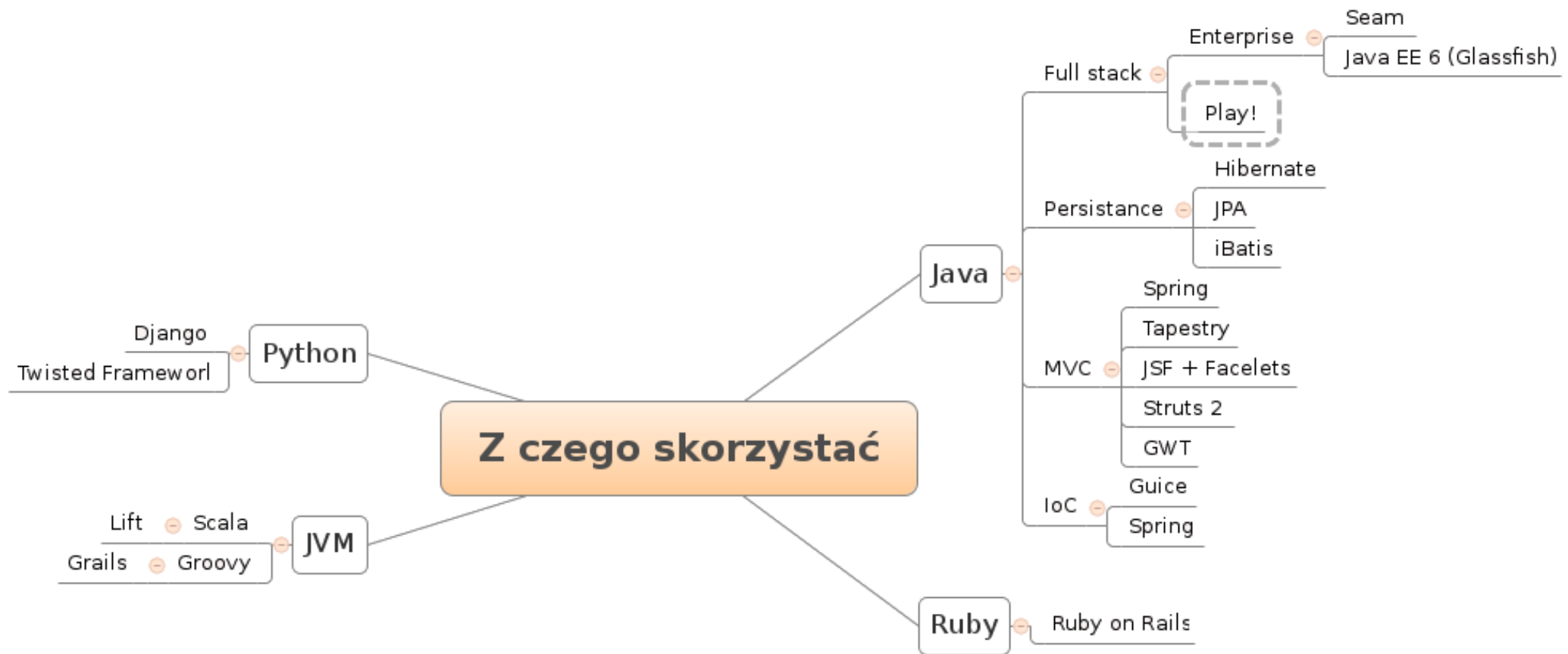
```
public void messages(int page) {  
    User connectedUser = User.find("byEmail", connected());  
    List<Message> messages = Message.find(  
        "user = ? and read = false order by date desc",  
        connectedUser  
    ).from(page * 10).fetch(10);  
    render(connectedUser, messages);  
}
```

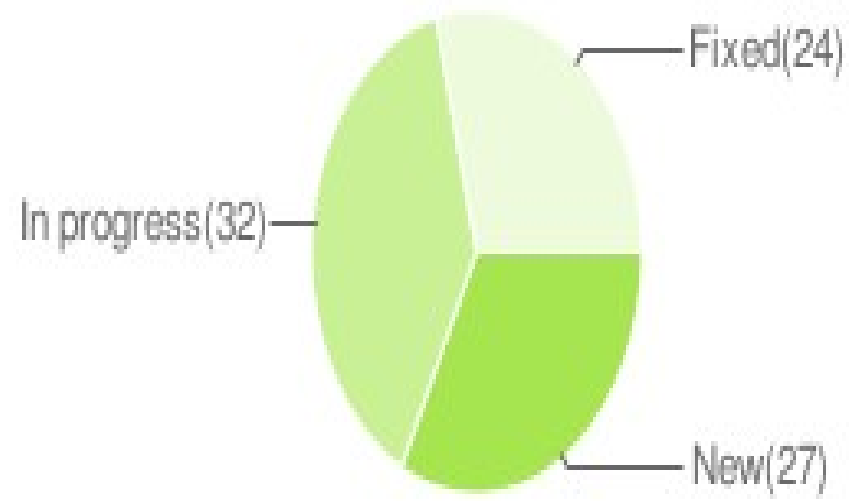


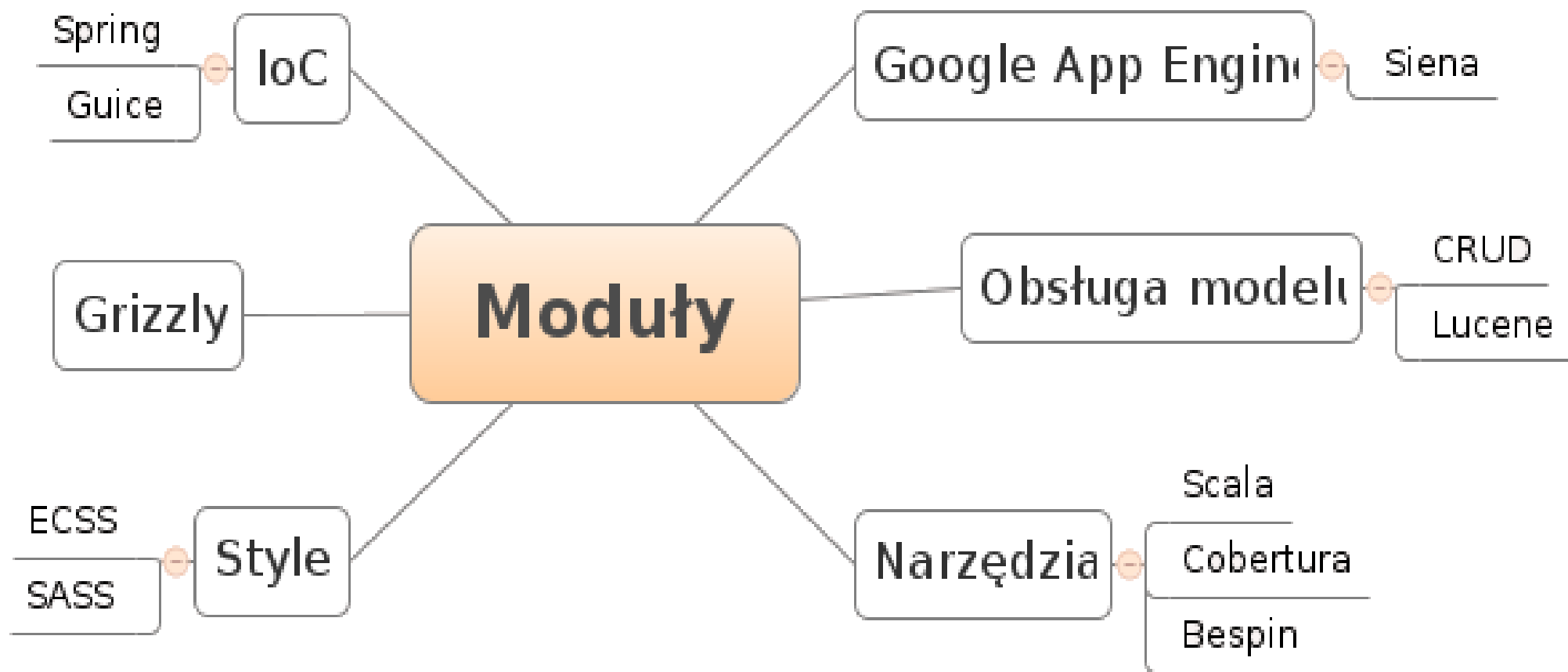
HTML czy JSON?

```
public void messages(int page) {  
    User connectedUser = User.find("byEmail", connected());  
    List<Message> messages = Message.find(  
        "user = ? and read = false order by date desc",  
        connectedUser  
    ).from(page * 10).fetch(10);  
    renderJSON(connectedUser, messages);  
}
```









Testowanie

```
public class ApplicationTest extends FunctionalTest {
```

```
    @Test
```

```
    public void testTheHomePage() {
```

```
        Response response = GET("/");
```

```
        assertStatus(200, response);
```

```
    }
```

```
}
```



Templates – Groovy baby!

```
{extends 'main.html' /}
```

```
{set title:'Beeing double test' /}
```

```
{stringParam} &{is}
```

```
{ifnot stringParam && stringParam.isDouble() }
```

```
<b>&{not}</b>
```

```
{/ifnot}
```

```
&{double}.
```

```
<br/>
```

```
Now just <a href="@{Show.application(1)}">click me</a>.
```



