

# MVC (JSR-371)

**BACK** ←  
**TO THE SERVER**

Florian Hirsch (dwins GmbH)

[github.com/lefloh](https://github.com/lefloh) – [twitter.com/lefloh](https://twitter.com/lefloh)

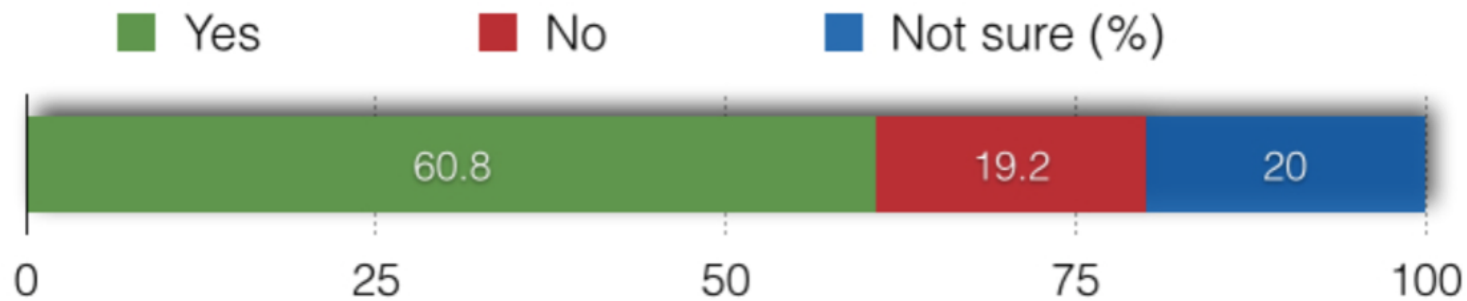
dwins  
DIGITAL WINS

**Why?**

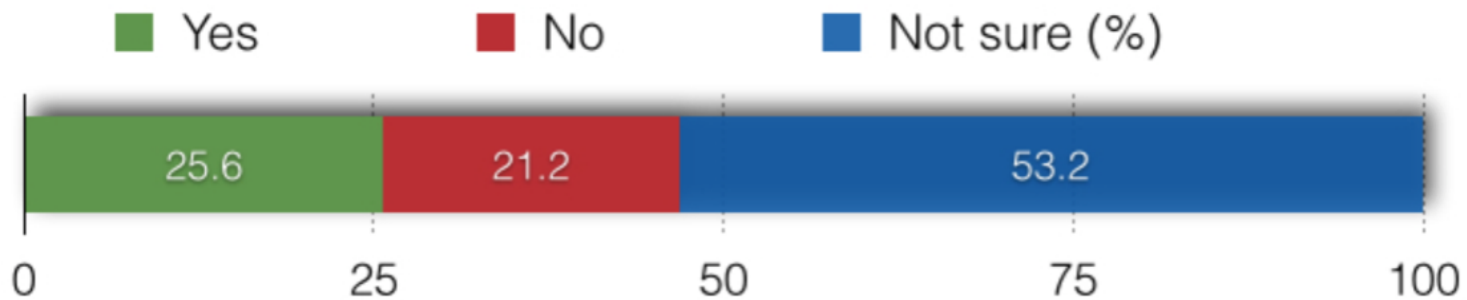
# Results from the Java EE 8 Community Survey

## MVC Support

Should Java EE provide support for MVC, alongside JSF?



Is there any one de-facto standard technology in this space to which we should look for inspiration?



# Current Java EE Options?

**Servlets & JSPs?**

JSF?

**Portlets?**

**JAX-RS?**



# ROCA?

Resource-oriented client architecture

[roca-style.org](http://roca-style.org)

# Goals

- Leverage existing Java EE technologies
- Integrate with CDI and Bean Validation
- Define a solid core to build MVC applications without necessarily supporting all the features in its first version
- Explore layering on top of JAX-RS for the purpose of re-using its matching and binding layers
- Provide built-in support for JSPs and Facelets view languages

# Non Goals

- Define a new view (template) language and processor
- Support standalone implementations of MVC running outside of Java EE
- Support REST services not based on JAX-RS
- Provide built-in support for view languages that are not part of Java EE

# Ozark View Engines

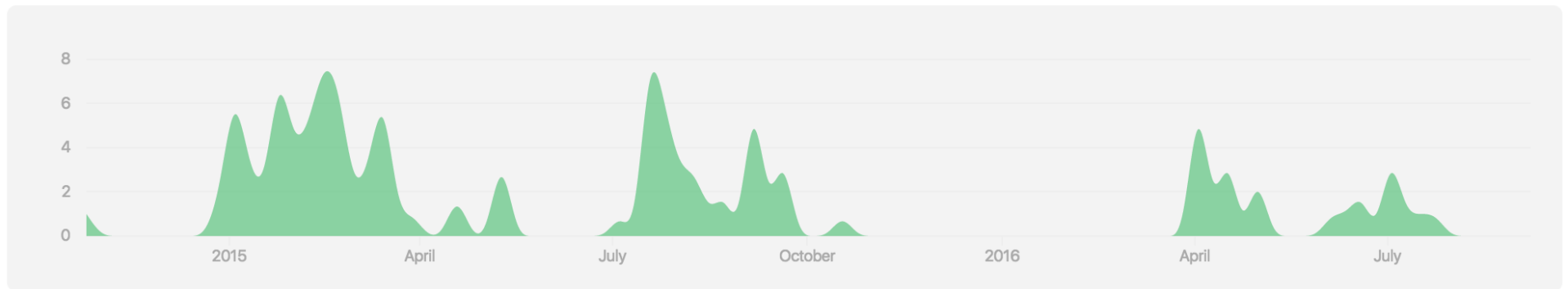
- AsciiDoc
- Freemarker
- Handlebars
- Jade
- JSR-223
- Mustache
- StringTemplate
- Thymeleaf
- Velocity
- (React)

**Status?**

# Contributions (Spec + Ozark)

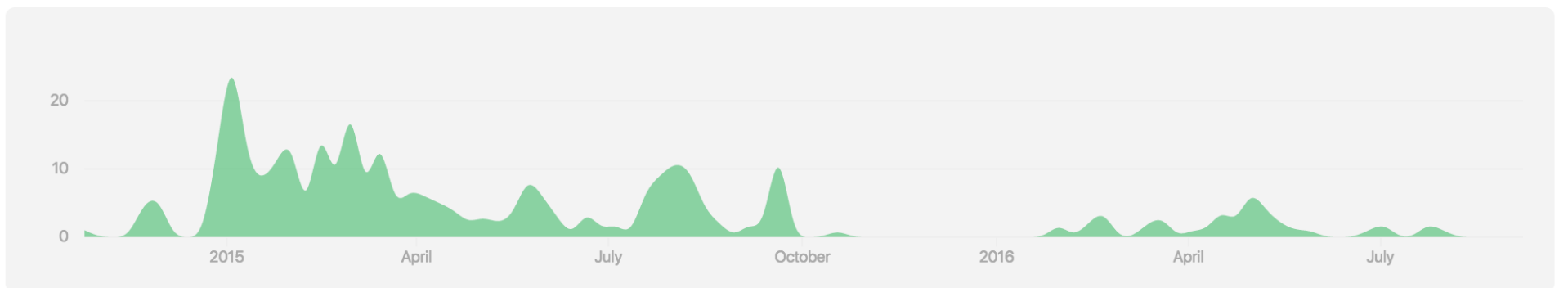
Nov 2, 2014 – Sep 14, 2016

Contributions to master, excluding merge commits



Nov 2, 2014 – Sep 14, 2016

Contributions to master, excluding merge commits



# Hands on!

Thanks to [Libor Kramoliš](#)

# Redirects

```
@RedirectScoped
public class Messages implements Serializable {

    ...

}

public String get() {
    return "redirect:/todos/" + id;
}
```



# Events

```
@ApplicationScoped
public class EventObserver {

    private static final Logger LOG =
        LoggerFactory.getLogger(EventObserver.class);

    public void beforeCtrl(@Observes BeforeControllerEvent e) {
        LOG.info("Resource {} will be handled by {}#{}",
            e.getUriInfo().getRequestUri(),
            e.getResourceInfo().getResourceClass().getName(),
            e.getResourceInfo().getResourceMethod().getName());
    }
}
```

# CSRF Protection

```
@POST
@CsrfValid
public String post(@RequestParam Todo todo) {
    ...
}

<input type="hidden"
       name="{{mvc.csrf.name}}" value="{{mvc.csrf.token}}" />
```

# i18n / l10n

```
public interface LocaleResolver {  
    Locale resolveLocale(LocaleResolverContext context);  
}
```

# i18n / l10n

```
@RequestParam("dob")  
@ConvertWith(DateConverter.class)  
private LocalDate dob;
```

# MvcUriBuilder

```
@Controller
public class TodoController {

    @GET
    @Path("todo/{id}")
    @UriRef("todo")
    public String getOne(@PathParam("id") long id) {
        ...
    }
}

{{ mvc.uri('todo', [4711] )}}
{{ mvc.uri('todo', {id: 4711} )}}
{{ mvc.uri('TodoController#getOne', {id: 4711} )}}
```

**Thanks**