JakartaEE Tips

Filters & Interceptors

JAKARTA EE

JAKARTA EE

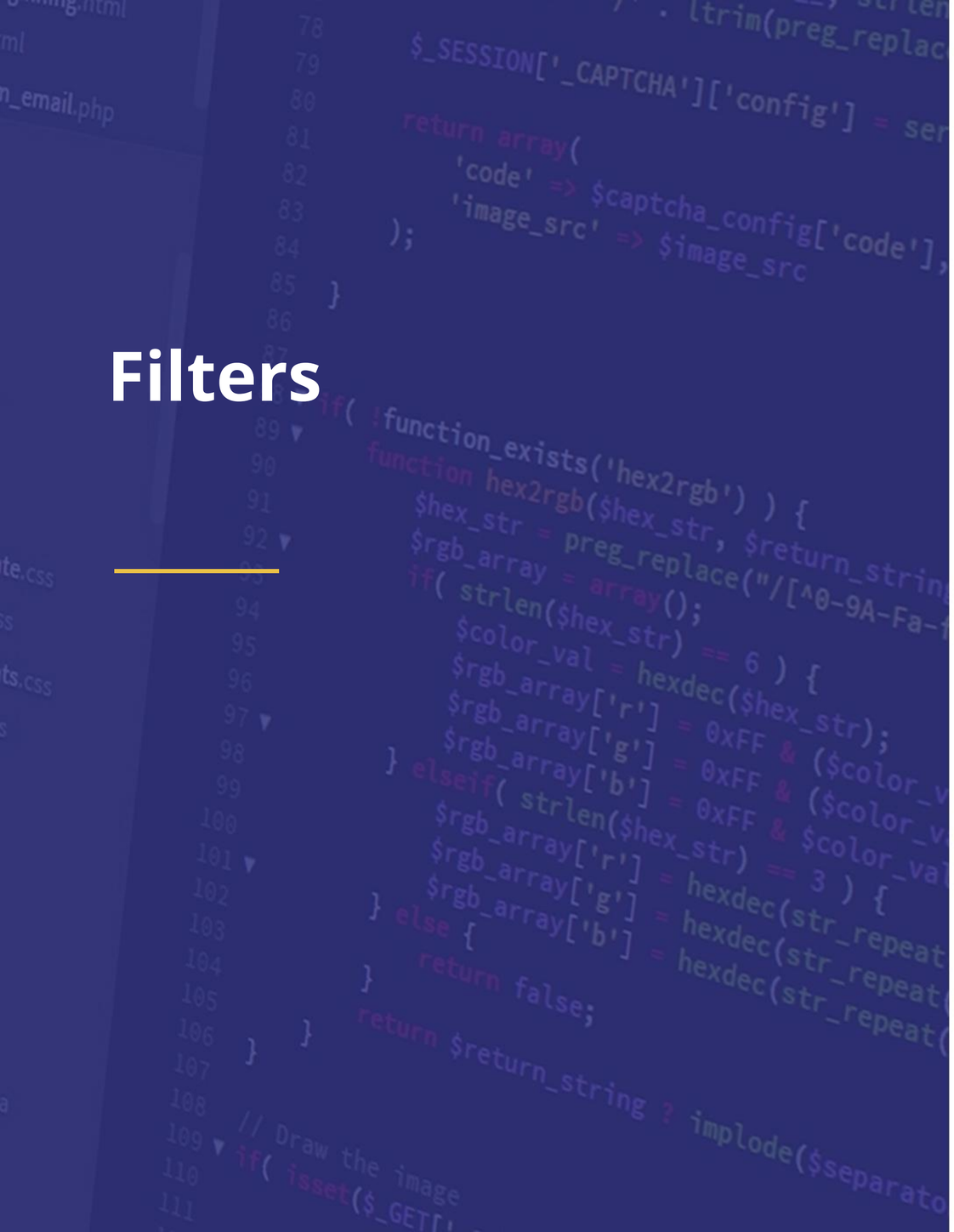
Jorge Cajas

- Consultant at MangoChango, S.A.
- Guate-JUG Leader
- JConf Guatemala Organizer
- +8 years of experience
- OCA certified
- Speaker since 2012

Table of contents

DRY - Don’t Repeat Yourself

- Filters
  - RequestFilters
  - ResponseFilters
- Interceptors
  - ReadInterceptors
  - WriteInterceptors
- JPA Listeners



Filters

JAKARTA EE

Always executed:
Does not matter if the resource exist, the filter is always executed

Can modify Properties
You can modify request or response properties like HTTP Headers, but not the body.

Where?:
Client or server side.

Types
PreMatching / PostMatching

Filters

PreMatching

Executed before searching the resource.

You can modify the URL of the resource, like an internal redirect.

You can modify HTTP Header values.

You can't’ modify the body content.

Needs to be annotated with @PreMatching

Filters

PostMaching

Executed after searching for the resource, does not matter if it exists or not.

You can't modify the URL for the resource.

You can modify HTTP Header values.

You can’t modify the body content.

This is the default behavior of filters.

Filters

```java
@Provider
@PreMatching
public class RedirectFilter implements ContainerRequestFilter {

  @Override
  public void filter(ContainerRequestContext reqContext) throws IOException {

      // redirect conditionally
      if (shouldRedirect(reqContext)) {
          reqContext.setRequestUri(URI.create("/temp"));
      }
  }

  private boolean shouldRedirect(ContainerRequestContext reqContext) {
      //todo it should be conditional return
      return true;
  }
}
```

Filters

Conditional Execution

```
@Secured
@Provider
@Priority(Priorities.AUTHENTICATION)
public class AuthFilter implements ContainerRequestFilter {


    @Override
    public void filter(ContainerRequestContext requestContext) throws IOException {

      String token = requestContext.getHeaderString("Authorization");
      User user = .... //Search for user
      if(user == null) trhow new Exceception("No autorizado")

    }
}
```

Filters

Conditional Execution

```
@NameBinding
@Retention(RetentionPolicy.RUNTIME)
@Target({ElementType.METHOD, ElementType.TYPE})
public @interface Secured {
}
```

```
@Secured
@Path("/users")
public class UsersController {

  @GET
  public List<User> users(){
    ///magic here
  }
}
```

Interceptors

JAKARTA EE

Not always executed
They are only executed if the resource is found

Can modify the body content
You can modify the request or response body content, also the HTTP Headers

Where?
Client and Server side.

Types
ReadInterceptor / WriteInterceptor

Interceptors

ReadInterceptor

```java
@Provider
public class RequestServerReaderInterceptor implements ReaderInterceptor {

    @Override
    public Object aroundReadFrom(ReaderInterceptorContext context) throws IOException {

      InputStream is = context.getInputStream();
      String body = new BufferedReader(new InputStreamReader(is)).lines()
                        .collect(Collectors.joining("\n"));

        context.setInputStream(new ByteArrayInputStream(
          (body + " message added in server reader interceptor").getBytes()));

        return context.proceed();
    }
}
```

JAKARTA EE

Interceptors

WriteInterceptor

```java
@Provider
public class RequestClientWriterInterceptor implements WriterInterceptor {

    @Override
    public void aroundWriteTo(WriterInterceptorContext context) throws IOException {

      Object entity = context.getEntity();
      StandardResponse response = new StandardResponse();
      response.setData(entity);
      response.setSuccess(true);
      response.setTime(LocalDateTime.now())

      context.setEntity(resoponse)

        context.proceed();
    }
}
```

Exception Mappers

JAKARTA EE

Intercepts an Exception by type

You can modify the response sent to the client for a better error handling

JPA Listeners

JAKARTA EE

Se ejecutan en el ciclo de vida de una entidad

@PrePersist / @PostPersist

@PreUpdate / @PostUpdate

@PreDelete / @PostDelete

JPA Listeners

JAKARTA EE

```java
@RequestScoped
public class BaseModelListener {

    @PrePersist
    private void onPersist(BaseModel model) {
        model.setCreatedAt(LocalDateTime.now());
        model.setCreatedBy(id);
    }


    @PreUpdate
    private void onUpdate(BaseModel model) {
       model.setUpdatedAt(LocalDateTime.now());
    }

    @PreRemove
    private void onRemove(BaseModel model) {
        model.setDeletedAt(LocalDateTime.now());
    }

}
```

JPA Listeners

JAKARTA EE

```java
@RequestScoped
public class BaseModelListener {

    @PrePersist
    private void onPersist(BaseModel model) {
        model.setCreatedAt(LocalDateTime.now());
        model.setCreatedBy(id);
    }


    @PreUpdate
    private void onUpdate(BaseModel model) {
       model.setUpdatedAt(LocalDateTime.now());
    }

    @PreRemove
    private void onRemove(BaseModel model) {
        model.setDeletedAt(LocalDateTime.now());
    }

}
```

JPA Listeners

```java
@MappedSuperclass
@EntityListeners(BaseModelListener.class)
public class BaseModel {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(nullable = false)
    private LocalDateTime createdAt;

    @Column
    private LocalDateTime updatedAt;

    @Column
    private LocalDateTime deletedAt;

}
```

Demo

JAKARTA EE

¿Questions?

JAKARTA EE

¡Gracias!

Jorge Cajas
@cajasmota
MangoChango, S.A.

JAKARTA EE