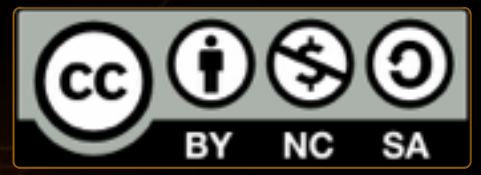
REST Controllers and Routing

Controllers, Routes, Responses



Borislav Yordanov
Technical Trainers
Software University
http://softuni.bg

Table of Contents

Symfony

Have a question?

SOFTWARE UNIVERSITY FOUNDATION

#sli.do

#2625

SOFTWARE UNIVERSITY FOUNDATION

Controller

Control, Control, You Must Learn Control

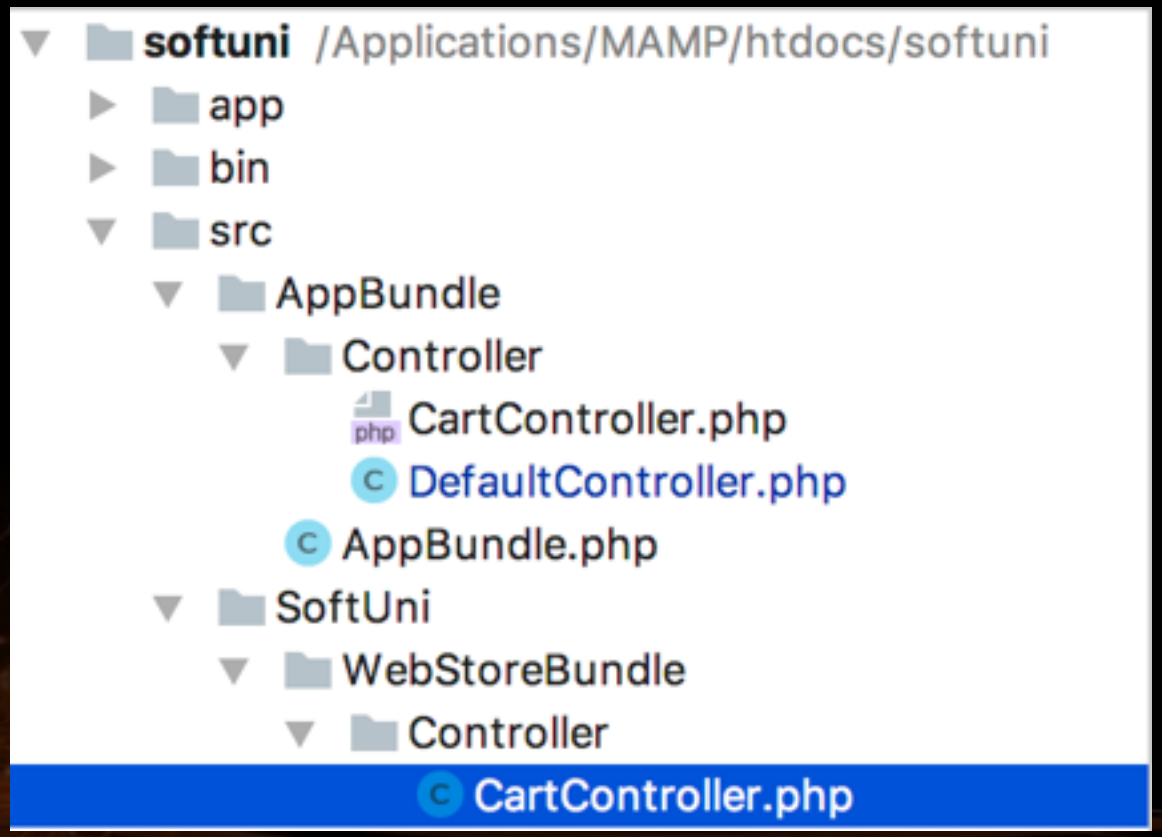
Controller

- Only concerned with *Controlling*
- Not responsible for the *doing*
- Takes care of web specific stuff

like authentication, cookies
request variables, query string

Controller

- Location: src/SomeNameBundle/Controller directory



Controller

- Controller names should end with "Controller"
- Controller action names should end with "Action"

```
class CartController extends Controller
{
    public function overviewAction(){}
    public function addAction(){}
    public function removeAction(){}
    public function emptyAction(){}
}
```

Controller

- Must always return Response:

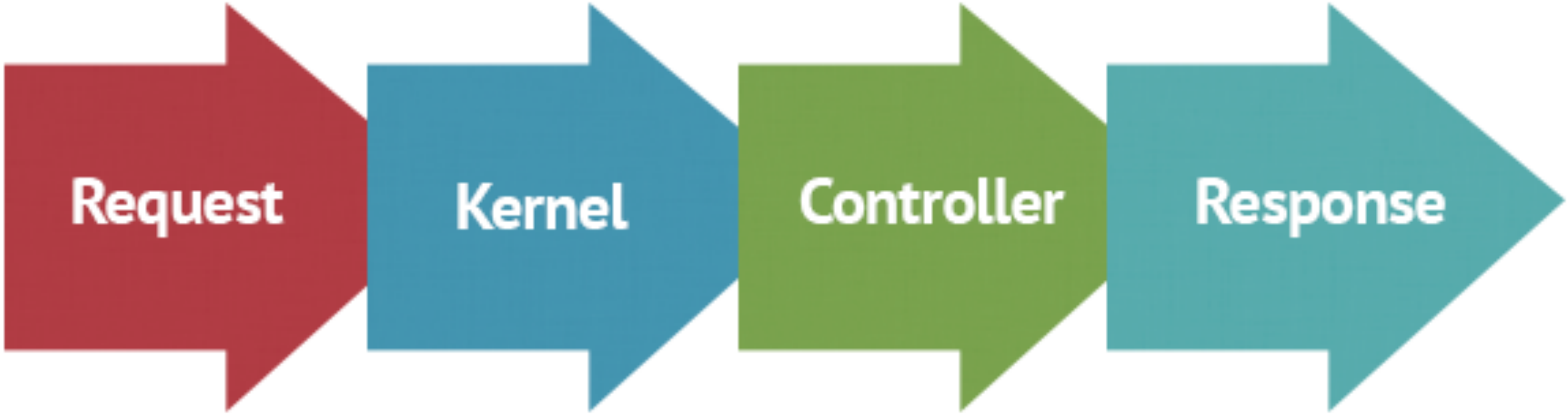
```
class CartController extends Controller
{
    public function overviewAction(){
        return $this->json(['num_items' => 2, 'total' => 10]);
    }
}
```

```
class CartController extends Controller
{
    public function overviewAction(){
        return new JsonResponse(['num_items' => 2, 'total' => 10]);
    }
}
```

- 5-10-20 rule:
  - 5 or less class variables
  - 10 or less actions
  - 20 or less lines of code for each action

Controller

Requests, Controller, Response Lifecycle



Controller

Symfony's Base Controller

- Provides access to helper methods and the service container
- Won't change your controller workflow

```
use Symfony\Bundle\FrameworkBundle\Controller\Controller;

class DefaultController extends Controller
{
```

Controller

SOFTWARE UNIVERSITY FOUNDATION

Symfony's Base Controller - helper methods example

```
public function helperMethodsAction()    {

    $this->generateUrl($route = 'homepage');
    $this->redirect($url = 'http://softuni.bg');
    $this->json($array = array());
    $this->render('AppBundle:Default:index.html.twig');
}
```

Symfony's Base Controller - services example

```
public function servicesAction()    {

    $templating = $this->get('templating');

    $router = $this->get('router');

    $mailer = $this->get('mailer');

    $doctrine = $this->get('doctrine');

}
```

Controller

Example

```php
<?php

namespace SoftUni\WebStoreBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;

class CartController extends Controller
{
    public function indexAction()
    {

        $items = $this->get('cart')->getItems();

        return $this->render('SoftUniWebStoreBundle:Cart:index.html.twig',
            array('items' => $items)
        );

    }
}
```

Problem 1: Create New Controller

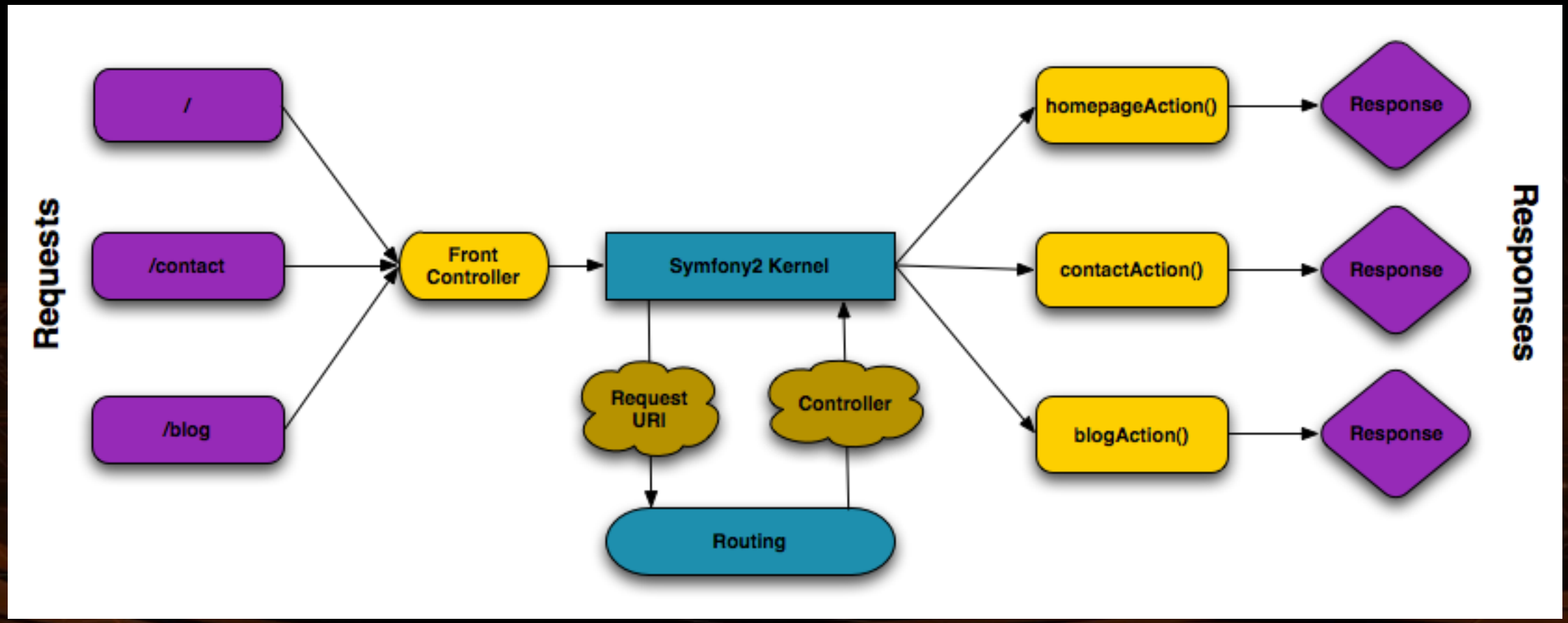
SOFTWARE UNIVERSITY FOUNDATION

Routing

Routes and Mapping to Controller

Routing

Process of breaking the URL into components and deciding what script to call



Routing

Four ways to define routes:

- Annotations
- YAML
- XML
- PHP

No difference in execution time

Only one type of configuration is allowed

Routing

Annotation: @Route

```php
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;

class CartController extends Controller
{
    private $_items = [
        ['title' => "PHP MVC Fundamentals", 'price' => 10.00],
        ['title' => "Symfony Framework", 'price' => 15.00],
    ];
    /**
     * @Route("cart/overview", name="cart_overview")
     * @return \Symfony\Component\HttpFoundation\Response
     */
    public function indexAction()
    {
        return $this->json($this->_items);
    }
}
```

```yaml
#app/config/routing.yml
soft_uni_web_store:
    resource: "@SoftUniWebStoreBundle/Controller"
    type:     annotation
```

Routing

YAML

```php
class CartController extends Controller
{
    private $_items = [
        ['title' => "PHP MVC Fundamentals", 'price' => 10.00],
        ['title' => "Symfony Framework", 'price' => 15.00],
    ];

    public function indexAction()
    {
        return $this->json($this->_items);
    }
}
```

```yaml
#src/SoftUni/WebStoreBundle/Resources/config/routing.yml
soft_uni_web_store_cart_overview:
    path:     /cart/overview
    defaults: { _controller: SoftUniWebStoreBundle:Cart:index }
```

```yaml
#app/config/routing.yml
soft_uni_web_store:
    resource: "@SoftUniWebStoreBundle/Resources/config/routing.yml"
    prefix:   /
```

Routing

Configure different HTTP methods for the route

Annotation: @Method

```php
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;

class CartController extends Controller
{
    private $_items = [
        ['title' => "PHP MVC Fundamentals", 'price' => 10.00],
        ['title' => "Symfony Framework", 'price' => 15.00],
    ];
    /**
     * @Route("cart/overview", name="cart_overview")
     * @Method({"GET","POST"})
     * @return \Symfony\Component\HttpFoundation\JsonResponse
     */
    public function indexAction()
    {
        return $this->json($this->_items);
    }
}
```

Routing

Configure different HTTP methods for the route

YAML

```yaml
#src/SoftUni/WebStoreBundle/Resources/config/routing.yml
soft_uni_web_store_cart_overview:
    path:      /cart/overview
    defaults: { _controller: SoftUniWebStoreBundle:Cart:index }
    methods:  [GET, POST]
```

Routing

SOFTWARE UNIVERSITY FOUNDATION

Route Parameters Annotation

```
/**
 * @Route("cart/delete/{id}", name="cart_delete_item")
 */
public function deleteAction($id){
    $this->get('cart')->delete($id);
    //TODO
}
```

YAML

```
public function deleteAction($id){
    $this->get('cart')->delete($id);
    //TODO
}
```

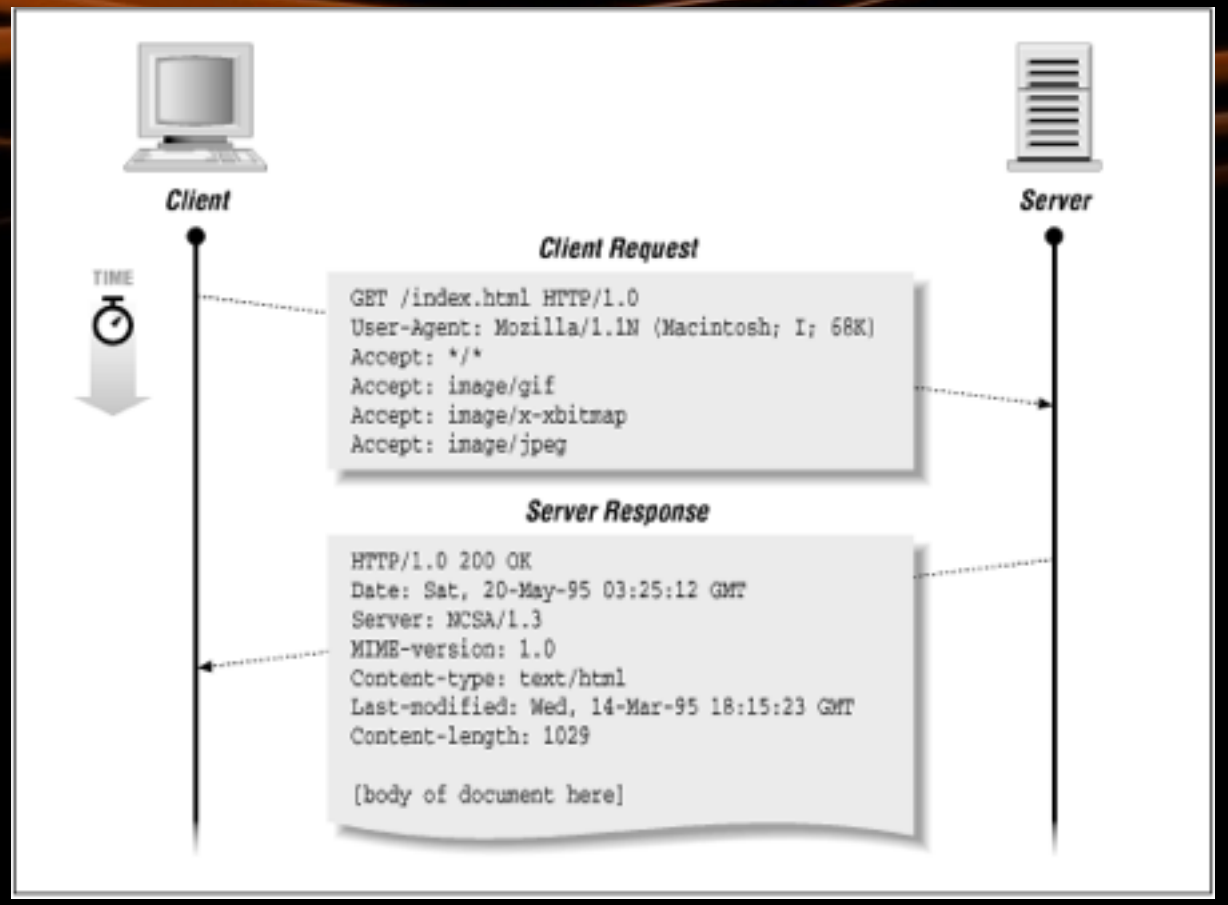
```
soft_uni_web_store_cart_delete_item:
    path:     /cart/delete/{id}
    defaults:     { _controller: SoftUniWebStoreBundle:Cart:delete }
```

Routing

Route Parameters

- Additional parameters are converted to query parameters
- Parameters names passed to the controller must match route parameters names
- Parameters and route parameters names must be the same

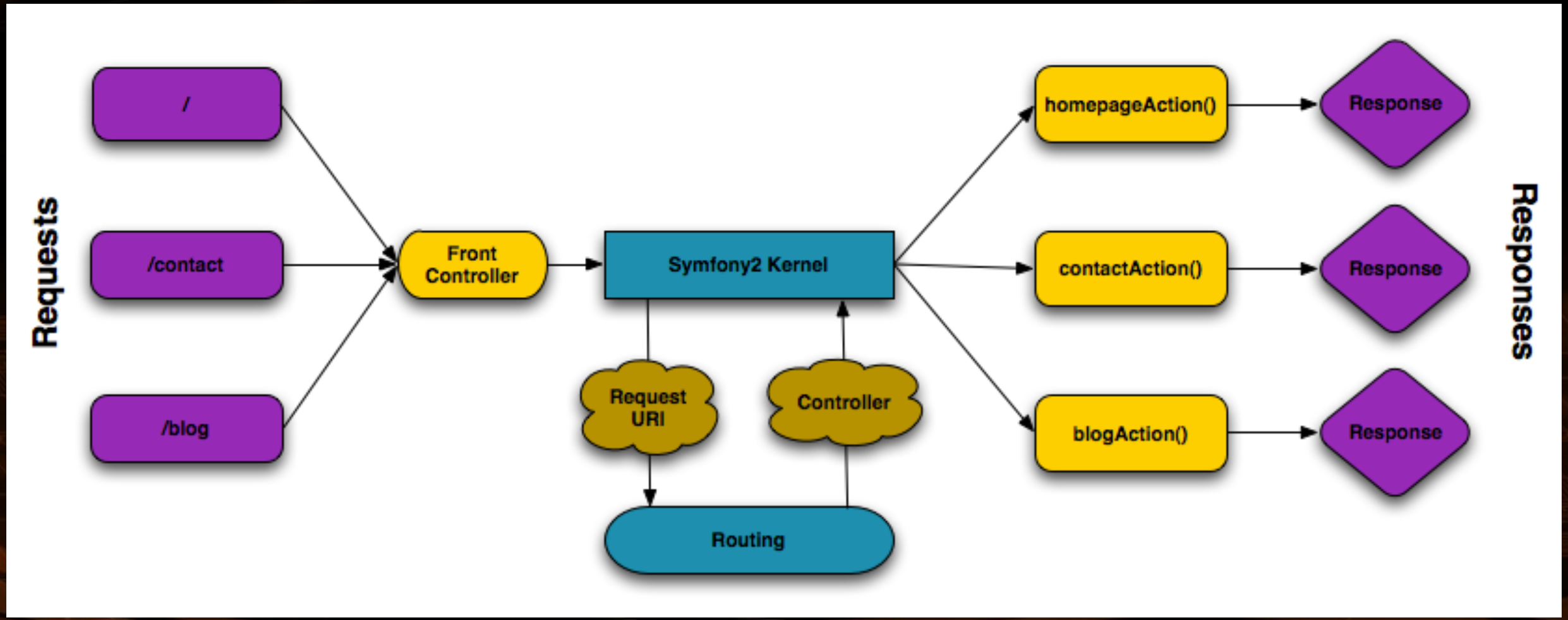
```
/**
 * @Route("cart/compare/{first_item}/{second_item}", name="cart_compare_items")
 */
public function compareAction($first_item, $second_item)  {}
```



Request & Response

Request & Response

Symfony Application Lifecycle



Request & Response

SOFTWARE UNIVERSITY FOUNDATION

Symfony Application Lifecycle

- Client sends an HTTP Request
- For each request the front controller is executed: app.php or app_dev.php
- Router component matches the request and execute controller
- Response object is returned

Request & Response

SOFTWARE UNIVERSITY FOUNDATION

Request

- Import the Request Object in controller class

```
use Symfony\Component\HttpFoundation\Request;
```

- To use it in your controller type-hint it with the Request class

```
/**
 * @Route("cart/overview", name="cart_overview")
 * @Method({"GET","POST"})
 */
public function indexAction(Request $request)
{
    $ref = $request->query->get('_ref');
    $session = $request->getSession();
}
```

Request & Response

SOFTWARE UNIVERSITY FOUNDATION

Request Examples:

```php
public function indexAction(Request $request)
{
    //get $_POST parameters
    $httpPostParam = $request->request->get('post_param');
    //get $_GET parameters
    $httpGetParam = $request->query->get('get_param');
    //get session
    $session = $request->getSession();
    //get cookies
    $cookies = $request->cookies->get('cookie_name');
    //get uploaded files
    $files = $request->files;
    //get request method
    $method = $request->getMethod();
```

Request & Response

SOFTWARE UNIVERSITY FOUNDATION

Response

- Import the Response Object in controller class

```
use Symfony\Component\HttpFoundation\Response;
```

```
public function helloAction(Request $request, $name) {
    $content = "Hello, $name";
    return new Response($content);
}
```

```
public function helloAction(Request $request, $name) {
    $content = "Hello, $name";
    return new Response($content,
                Response::HTTP_OK,
                array('content-type' => 'text/html'));
}
```

Request & Response

JsonResponse

```
use Symfony\Component\HttpFoundation\JsonResponse;
```

```
class CartController extends Controller
{
    private $_items = [
        ['title' => "PHP MVC Fundamentals", 'price' => 10.00],
        ['title' => "Symfony Framework", 'price' => 15.00],
    ];

    /**
     * @Route("cart/summary", name="cart_summary")
     */
    public function cartSummaryAction()    {
        return new JsonResponse($this->_items);
    }
```

Request & Response

SOFTWARE UNIVERSITY FOUNDATION

Response Content-Types

- json - application/json
- xml - application/xml
- png image - image/png
- html - text/html
- full list of content types: developer.mozilla.org

Request & Response

Set Conten-type Examples

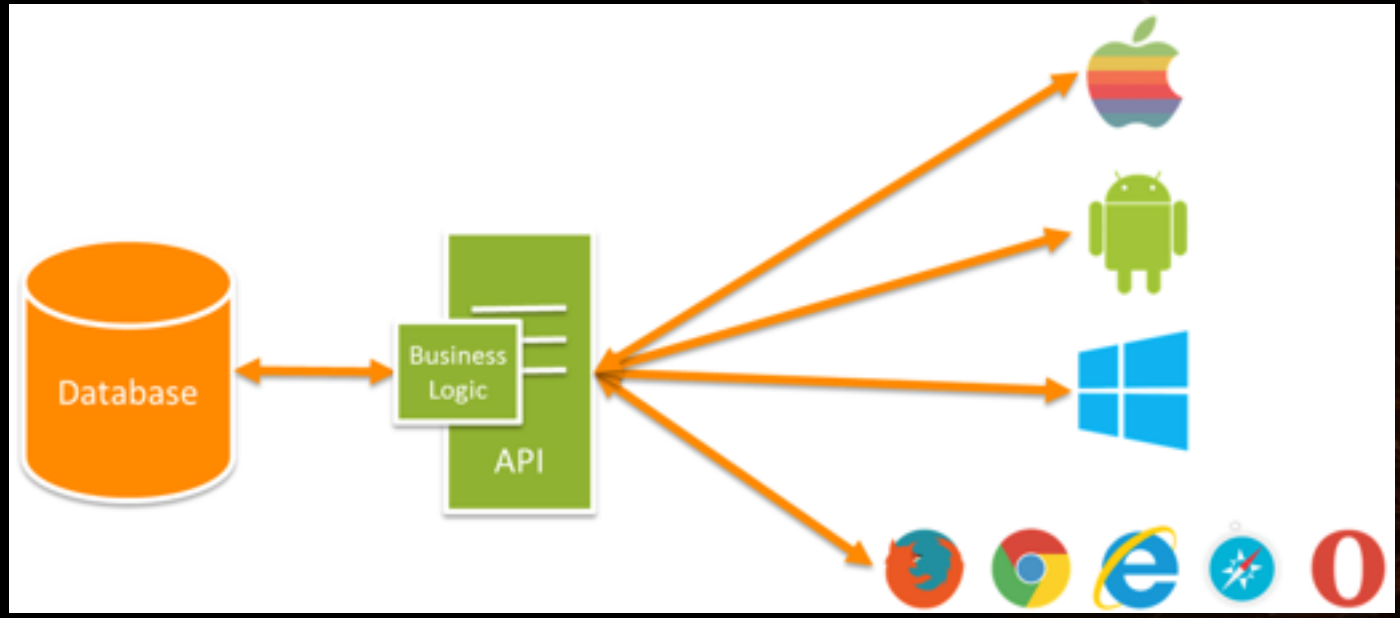
```
public function jsonResponseAction()    {
    return new Response(
        json_encode($this->_items),
        Response::HTTP_OK,
        array('application/json')
    );
}
```

```
public function jsonResponseAction()    {
    $response = new Response();
    $response->setContent('<strong>Hello World</strong>');
    // the headers public attribute
    // is a Symfony\Component\HttpFoundation\ResponseHeaderBag
    $response->headers->set('Content-Type', 'text/html');
    $response->setStatusCode(Response::HTTP_NOT_FOUND);
    return $response;
}
```

REST

What is REST?

- REpresentational State Transfer
- Interface from one program to another
- REST is stateless
- REST is resource based



REST

REST & HTTP

- Request methods:
  - GET - Get resource
  - POST - Create resource
  - PUT - Update resource
  - PATCH - Partly update resource
  - DELETE - Delete resource
  - HEAD - Get headers for resource
- Response: json, xml

REST

SOFTWARE UNIVERSITY FOUNDATION

REST URLs

- GET - HTTP://MYSITE.COM/API/USERS
- GET - HTTP://MYSITE.COM/API/USERS/1
- POST - HTTP://MYSITE.COM/API/USERS
- PUT - HTTP://MYSITE.COM/API/USERS/1
- DELETE - HTTP://MYSITE.COM/API/USERS/1

REST

REST Controllers

A Symfony controller configured for specific http method

```
class ItemController extends Controller
{
    /**
     * http://domain.com/item/1
     * @Route("item/{id}", name="api_get_item")
     * @Method({"GET"})
     */
    public function getItemAction($id)    {
        return new JsonResponse($this->_items[$id]);
    }
```

REST

REST Controllers

Add prefix to your controller using the @Route annotation

```
/**
 * @Route("/api/v1/")
 */
class ItemController extends Controller
{
    /**
     * http://domain.com/api/v1/item/1
     * @Route("item/{id}", name="api_get_item")
     * @Method({"GET"})
     */
    public function getItemAction($id)    {
        return new JsonResponse($this->_items[$id]);
    }
```

REST

REST Controllers

Prefix with YAML

```
#app/config/routing.yml
soft_uni_web_store:
    resource: "@SoftUniWebStoreBundle/Resources/config/routing.yml"
    prefix:   /api/v1/
```

Summary

- Controllers - Create controllers, actions, 5-10-20 rule
- Routing - Define routes, configuration, route parameters
- Request & Response - Request methods, Response types
- REST - Rest concepts, REST URIs, Responses

REST Controllers and Routing

SOFTWARE UNIVERSITY FOUNDATION

Questions?

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg
- Software University Foundation
  - softuni.org
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

SOFTWARE UNIVERSITY

SOFTWARE UNIVERSITY FOUNDATION

FORUM

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg
- Software University Foundation
  - softuni.org
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg