



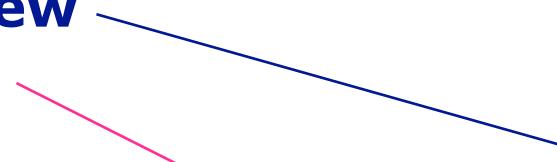
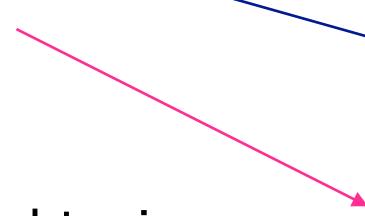
Distance Learning System



MVC Šablon

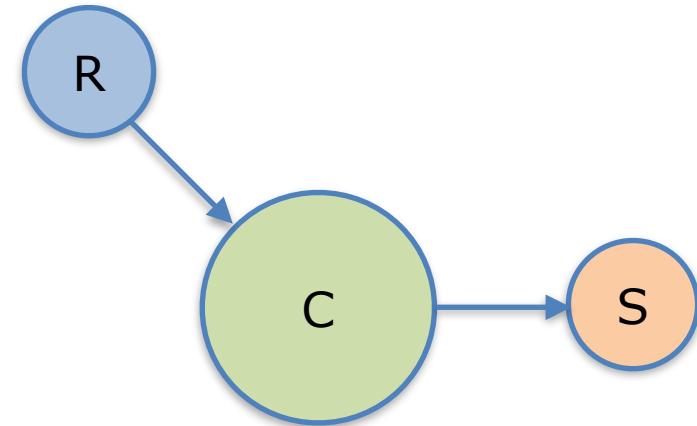
Web Application Development

MVC Struktura

- MVC je dizajn šablon u kome se aplikacija deli na tri celine:
 - Funkcionalnost - **Controller** 
 - Prezentaciju - **View** 
 - Podatke - **Model** 
- Zahvaljujući ovoj strukturi:
 - Aplikacija se lakše ažurira
 - Aplikacija se lakše testira
 - Aplikacija se bolje kreira u distribuiranom okruženju

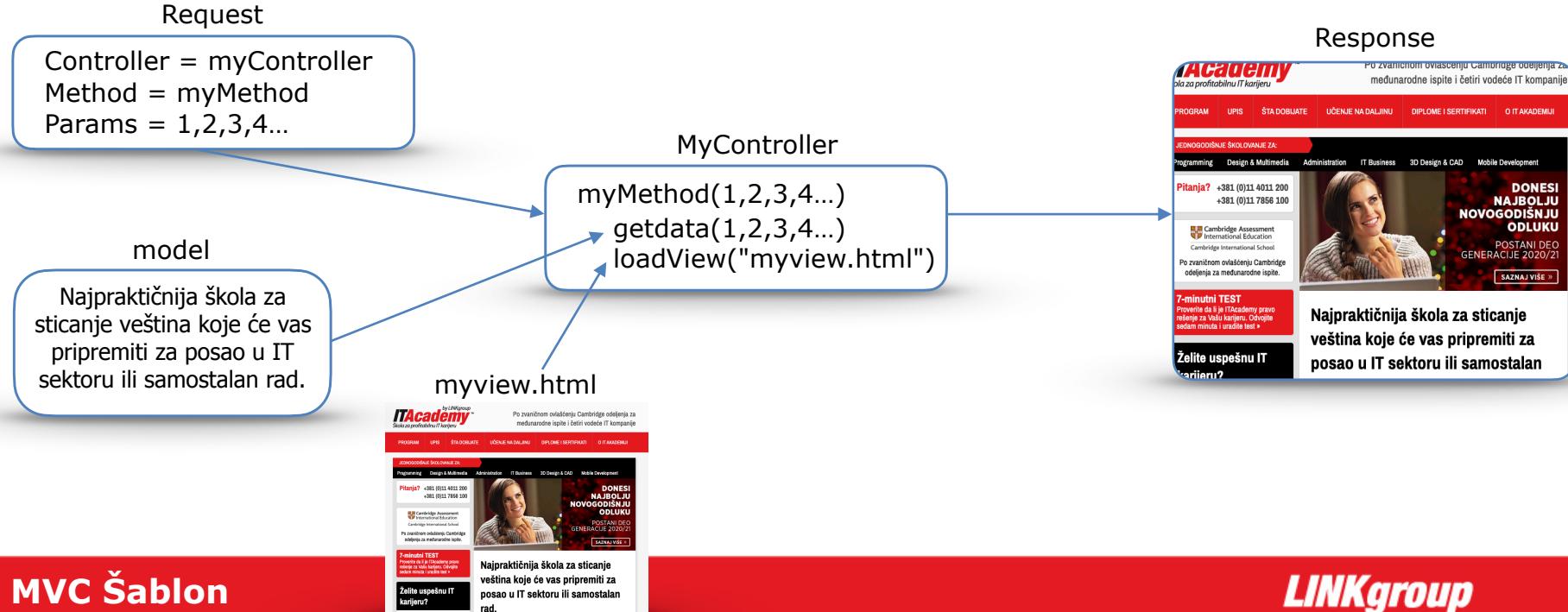
Ostali akteri u MVC-u

- Iako je MVC predstavljen kroz tri komponente (Model, View i Controller) u njegovoj implementaciji najčešće ima još učesnika
 - Sistem rutiranja
 - Servisi
 - OR mapiranje



MVC web - životni ciklus

- Kod web aplikacija, mvc ciklus počinje HTTP zahtevom
- Zahtev treba da sadrži dovoljno informacija za dalje procesiranje



MVC implementacija

(wab-ex04 mymvc)

- U osnovi, za MVC implementaciju, neophodno je imati dispatcher i jak objektni model

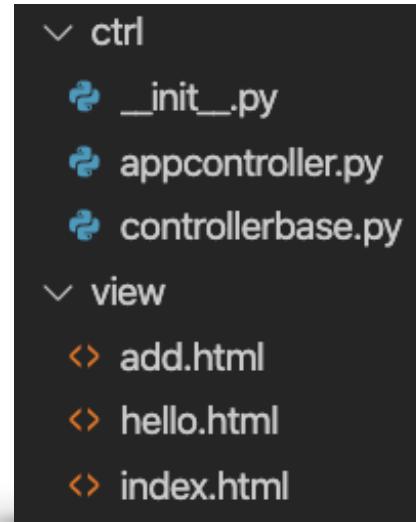
Dispatcher jer ulazna tačka. On presreće zahteve i aktivira kontrolere i njihove metode

```
class MyHandler(srv.SimpleHTTPRequestHandler):  
    def do_GET(self):  
        parts = parse.urlparse(self.path)  
        self.params = dict(parse.parse_qsl(parts.query))  
        if "ctrl" in self.params:  
            ctrlname = self.params["ctrl"] + "controller"  
            mod = importlib.import_module("ctrl."+ctrlname,"ctrl")  
            ctrlclass = getattr(mod,ctrlname)  
        else:  
            ctrlclass = ctrl.appcontroller.appcontroller  
        c = ctrlclass(self)  
        self.wfile.write("HTTP/1.1 200 Ok\r\n\r\n".encode("utf-8"))  
        if "mtd" in self.params:  
            mtd = getattr(c,self.params["mtd"])  
            mtd()  
        else:  
            c.index()
```

MVC implementacija - struktura direktorijuma

(wab-ex04 mymvc)

Pogledi i kontroleri treba
da budu tematski
grupisani



MVC implementacija - bazni kontroler

(wab-ex04 mymvc)

Bazni kontroler
treba da sadrži
generalnu logiku

```
class controllerbase:
    def __init__(self, request):
        self.request = request
    def index(self):
        self.loadView("index")
    def loadView(self, view, params={}):
        r = open(f"view/{view}.html")
        content = r.read()
        for k in params:
            content = content.replace("{{"+k+"}}", str(params[k]))
        self.request.wfile.write(content.encode("utf-8"))
        r.close()
```

MVC implementacija - kontroler

(wab-ex04 mymvc)

Deca kontroleri
sadrže specijalnu
logiku

```
import ctrl.controllerbase
class appcontroller(ctrl.controllerbase.controllerbase):
    def hello(self):
        self.loadView("hello")
    def add(self):
        a = self.request.params["a"]
        b = self.request.params["b"]
        c = int(a) + int(b)
        self.loadView("add", {"a":a,"b":b,"c":c})
```

MVC implementacija - pogled

(wab-ex04 mymvc)

Pogledi su šabloni za rezervisanim prostorima za dinamički sadržaj

```
<h1>Hello {{name}}</h1>
```