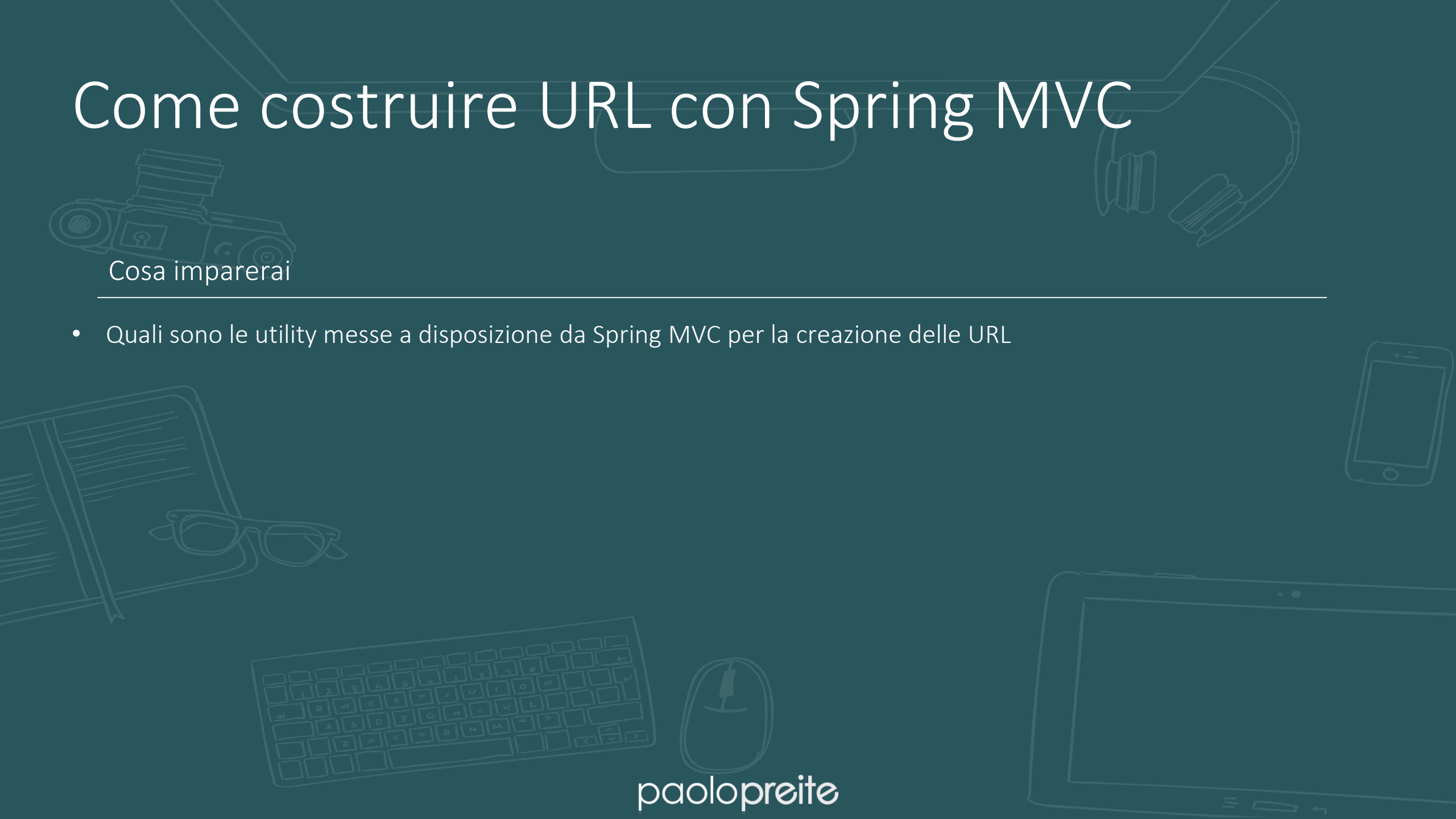
Come costruire URL con Spring MVC

Cosa imparerai

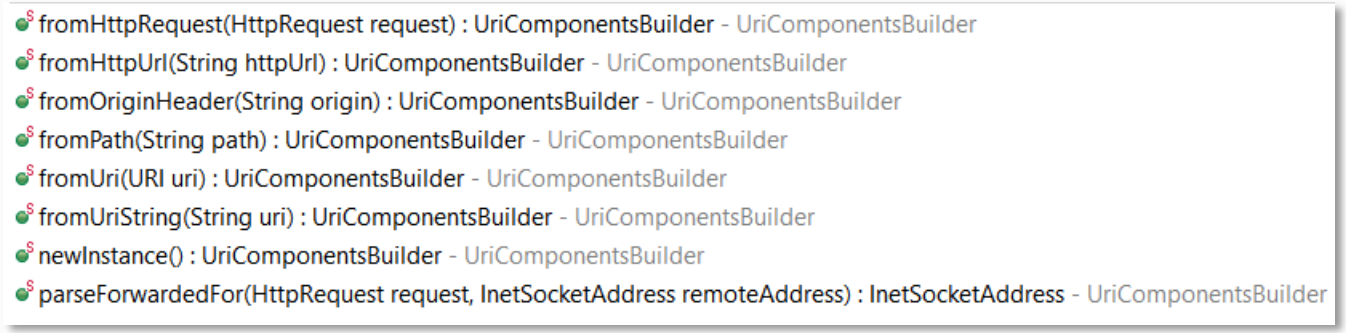
- Quali sono le utility messe a disposizione da Spring MVC per la creazione delle URL



Come costruire URL con Spring MVC

UriComponents e **UriComponentsBuilder**

UriComponentsBuilder (*package org.springframework.web.util*) è una classe che consente di creare URI utilizzando diversi metodi statici.



I metodi statici ritornano oggetti di tipo *UriComponentsBuilder* con i quali possiamo effettuare diverse operazioni, tra cui:

- ❑ Aggiungere parametri in query string
- ❑ Fare l'encoding dei parametri

paolopreite

Come costruire URL con Spring MVC

UriComponents e **UriComponentsBuilder**

Facciamo un esempio…

```
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromUriString("https://mysite.com/myservice/");
uriBuilder.queryParam("param1", "123<script>alert();</script>");
uriBuilder.queryParam("param2", "345");
uriBuilder.queryParam("param3", "asd");

UriComponents uriComponent = uriBuilder.encode().build();

URI uri = uriComponent.toUri();
```

```
https://mysite.com/myservice/?param1=123%3Cscript%3Ealert();%3C/script%3E&param2=345&param3=asd
```

- Il metodo **fromUriString** è utilizzato per creare un oggetto di tipo *UriComponentsBuilder* che abbia la URI indicata nella stringa passata come argomento al metodo;
- Il metodo **queryParam** è utilizzato per aggiungere un parametro (o più parametri) in query string;
- Il metodo **encode** effettua l'encoding di tutti i componenti della URI, effettuando l'escape di eventuali caratteri non-ASCII ed illegali
- Il metodo **build** crea un oggetto di tipo *UriComponents*, dal quale possiamo ottenere l'oggetto di tipo *URI* (*java.net.URI*)

paolopreite

Come costruire URL con Spring MVC

UriComponents e **UriComponentsBuilder**

Facciamo un altro esempio…

```
UriComponentsBuilder uriBuilder = UriComponentsBuilder.fromUriString("https://mysite.com/{service}/");
uriBuilder.queryParam("param1", "{val1}");
uriBuilder.queryParam("param2", "{val2}");
uriBuilder.queryParam("param3", "{val3}");

UriComponents uriComponent = uriBuilder.encode().build();

String val1 = "hello";
String val2 = "Paolo";
String val3 = "123";
String val4 = "asd";

URI uri = uriComponent.expand(val1, val2, val3, val4).toUri();
```

```
https://mysite.com/hello/?param1=Paolo&param2=123&param3=asd
```

- Nei metodi **fromUriString** e **queryParam** possiamo utilizzare la notazione *{nomevariabile}* per mettere un *placeholder* che verrà sostituito con delle stinghe nel metodo **expand(…)** della classe *UriComponents*;

paolopreite

Come costruire URL con Spring MVC

Costruire URI relative alla ServletRequest attuale

Se dobbiamo realizzare dei link interni possiamo utilizzare la classe **ServletUriComponentsBuilder**.

Questa classe ha diversi metodi statici per generare URI, ad esempio:

- *fromContextPath*: genera una URI che contiene il context path della web application (ad es. http://localhost:8080/hello-web)

- *fromRequest*: genera una URI che contiene il context path della web application ed il path del controller invocato nella request corrente (ad es. http://localhost:8080/hello-web/helloFreeMarker)

Anche in questa classe possiamo utilizzare i metodi per gestire e configurare le URI, ad es queryParam(), encode(), build(),…

paolopreite

Come costruire URL con Spring MVC

Costruire URI relative alla ServletRequest attuale

Facciamo un esempio…

```
UriComponents uriComponents = ServletUriComponentsBuilder.fromRequest(request)
        .queryParam("param1", "{val1}")
        .queryParam("param2", "{val2}")
        .queryParam("param3", "{val3}")
        .encode()
        .build();

URI uri = uriComponents.expand("123","123","123").toUri();
```

paolopreite

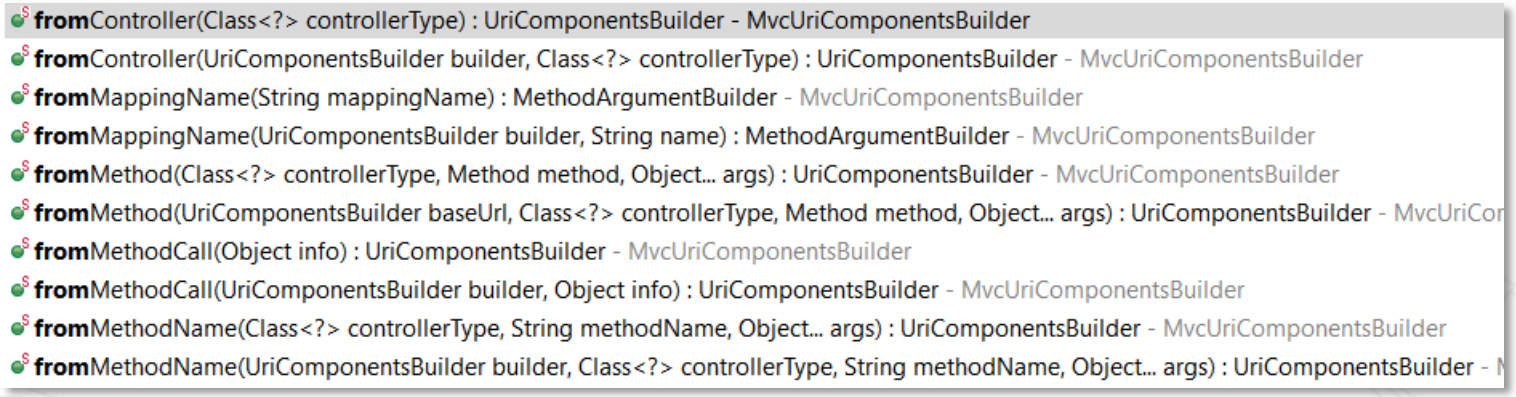
Come costruire URL con Spring MVC

Costruire Link ad un controller

Se vogliamo creare un link ad un controller, possiamo utilizzare la classe **MvcUriComponentsBuilder**.

In particolare, Spring consente di creare URI dinamiche attraverso la notazione *{path}* che può essere utilizzata nel controller o nel metodo (o in tutt'e due…)

Anche in questo caso abbiamo diversi metodi da utilizzare, tra cui *fromController()* che consente di specificare la classe controller a cui è indirizzata la richiesta.



paolopreite

Come costruire URL con Spring MVC

Costruire Link ad un controller

Facciamo un esempio…

Supponiamo di avere un controller che gestisce la visualizzazione di un catalogo prodotti, sulla base di una categoria ed abilita un set di funzionalità sulla base di un parametro (ad es. *view actions, view in list, view in table* ecc…)

Per fare questo possiamo creare:

- ❑ un controller che avrà come parametro dinamico ***{action}***
- ❑ un metodo che avrà come parametro dinamico ***{category}***

paolopreite

Come costruire URL con Spring MVC

Costruire Link ad un controller

```java
@Controller
@RequestMapping(path = "/{action}")
public class HelloController {
    @GetMapping(path = "/products/{category}")
    public String getProducts(@PathVariable Long category) {

        return "products";
    }

    @GetMapping(path = "/show")
    public String showProducts() {
        UriComponents uriComponents = MvcUriComponentsBuilder
                .fromMethodName(HelloController.class, "getProducts", 1231).buildAndExpand("viewlist");

        URI uri = uriComponents.encode().toUri();

        return "redirect:" + uri.toString();
    }
```

Attraverso la classe **MvcUriComponentsBuilder** creiamo un link al controller, indicando la classe del controller e passando i parametri che sostituiranno le variabili **{action}** e **{category}**.

Esempi di URI che possiamo gestire con questa tecnica sono:

http://localhost:8080/hello-web/viewlist/products/123

http://localhost:8080/hello-web/viewlist/products/456

http://localhost:8080/hello-web/viewlistactions/products/678

paolopreite

Di cosa abbiamo parlato in questa lezione

- Quali sono le utility messe a disposizione da Spring MVC per la creazione delle URL