Routing

Routing jest mechanizmem mapującym adresy URL do odpowiadającym im akcjom kontrolera. Kiedy użytkownik klika w link, przechodzi do strony poprzez bezpośrednie wpisanie adresu lub następuje automatyczne przekierowanie rozpoczyna się nowe połączenie z serwerem. W założeniach protokołu http adres URL zawierał w sobie ścieżkę reprezentującą fizyczne położenie pliku na dysku.

TODO: Obrazek z katalogami i adresem

Takie rozwiązanie, choć w pełni dające możliwość zbudowania kompletnej aplikacji, posiada szereg wad, m.in.:

* Eksponowanie klientowi struktury i hierarchii plików oraz katalogów na serwerze. O ile prawdą jest, że dla poprawnie zabezpieczonej aplikacji powinno być obojętne czy struktura plików jest jawna czy nie, to jednak jej znajomość przez hakera jeszcze przed atakiem pozwala na jego sprawniejsze i szybsze wykonanie.
* Trudność zaimplementowania dispachowania {TODO: język} akcji kontrolera.
* Bezpośredni dostęp poprzez ścieżkę narzuca problem ograniczania dostępu do konkretnych plików, które nie powinny być dostępne dla wszystkich użytkowników, bądź tylko dla wybranej grupy.
* Nieatrakcyjny i nieprzyjazny wygląd adresów URL dla użytkownika.

Mechanizm routingu umożliwia obejście tego problemu. Żądanie do serwera jest przechwytywane, a następnie kierowanie do warstwy routingu. Tam tzw. „Router” jest odpowiedzialny za przeanalizowanie żądania, nie tylko pod kątem podanego adresu, ale również użytej metody http.

TODO: przykład przechwytywania z użyciem .httpaccess

Czym jest route?

„Route” (z ang. „szlak” lub „trakt”) definiuje jeden lub grupę adresów URL. Określa również liczne parametry jakie charakteryzują dany adres. Mechanizm routingu po otrzymaniu do przetworzenia wysłanego żądania do serwera przegląda kolejno skonfigurowane routy. Poszukuje tak długo, aż uda mu się dopasować parametry żądania do któregoś ze skonfigurowanych routów. Jeśli nie uda się dopasować

Definiowanie routów

Wszystkie adresy URL, których dostarcza do użytku aplikacja MVC powinny być zdefiniowane. Istnieje wiele sposobów na definiowanie routów. Niektóre frameworki umożliwiają tworzenie zasad routingu na kilka sposobów, inne tylko jeden.

Do najpowszechniejszych sposobów definiowania routów należą:

* Definiowanie poprzez plik konfiguracyjny
* Definiowanie bezpośrednio poprzez kod programu
* Definiowanie poprzez adnotacje

Każdy sposób ma swoje wady i zalety, jednak przed porównaniem należy się głębiej z nimi zapoznać. Do porównania użyjemy przykładowego routa, który będzie zdefiniowany na różne sposoby.

Cechy routa:

* będzie odpowiedzialny do kierowania na stronę z listą samochodów
* samochody będą podzielone na kategorie
* zawiera paginację

Szablon adresu, którego chcemy użyć:

TODO: samochody/{category}/{page}

Powyższy szablon składa się z trzech sekcji:

* samochody – wymagamy, aby ta część adresu była niezmienna
* category – ta sekcja jest parametrem - dowolnym ciągiem znaków reprezentującym kategorię
* page – ta sekcja również jest parametrem - numerem strony, jest liczbą całkowitą, parametrem opcjonalnym (zakładamy, że jeśli nie zostanie podana to kierujemy na pierwszą stronę)

Przykładowe adresy spełniające powyższe wymogi:

samochody/terenowe/5

samochody/osobowe

samochody/ciężarowe/17

Chcemy, aby ten route kierował do określonej akcji kontrolera:

* kontroler: Cars
* akcja: List

Posiadając powyższe wymagania jakie ma spełnić route możemy przejść do porównania różnych sposobów jego definicji.

Definiowanie poprzez plik konfiguracyjny

sda

Definiowanie bezpośrednio poprzez kod programu

Definiowanie poprzez adnotacje

TODO: constrainty

Kontrolery

Model

TODO: opisać jak model w ujęciu ORM bywa mylony z modelem MVC

<http://michalorman.pl/blog/2010/03/model-widok-kontroler/>

Widok

Słownik:

Hacker

MVC

Model

Widok

Kontroler

Framework

Żądanie – Request

Paginacja