

Principy tvorby klienta pro webové/enterprise aplikace

Front-end technologie

Java-based

Servlet API

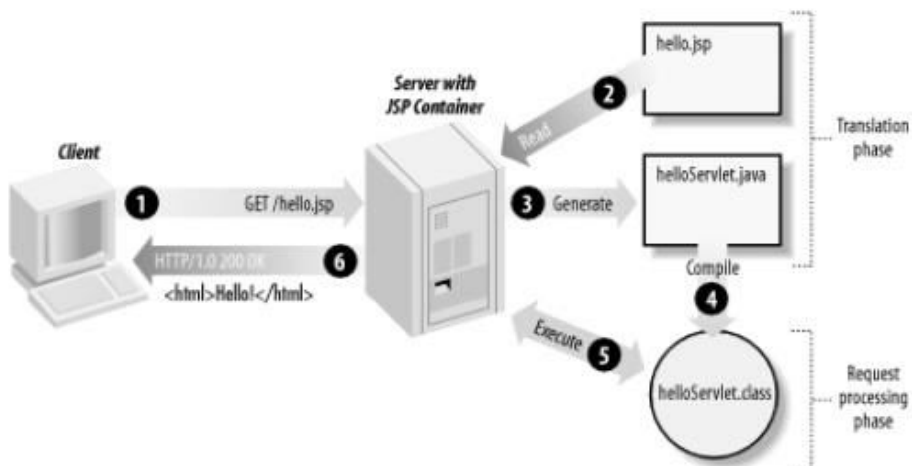
- (HTTP-specific) třídy pro zpracování request/response
- response psána přímo do output streamu poslaného klientovi
- může zpracovávat vícero požadavků současně

```
public class ServletDemo extends HttpServlet{

    public void doGet (HttpServletRequest request, HttpServletResponse
        response)
        throws IOException{
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

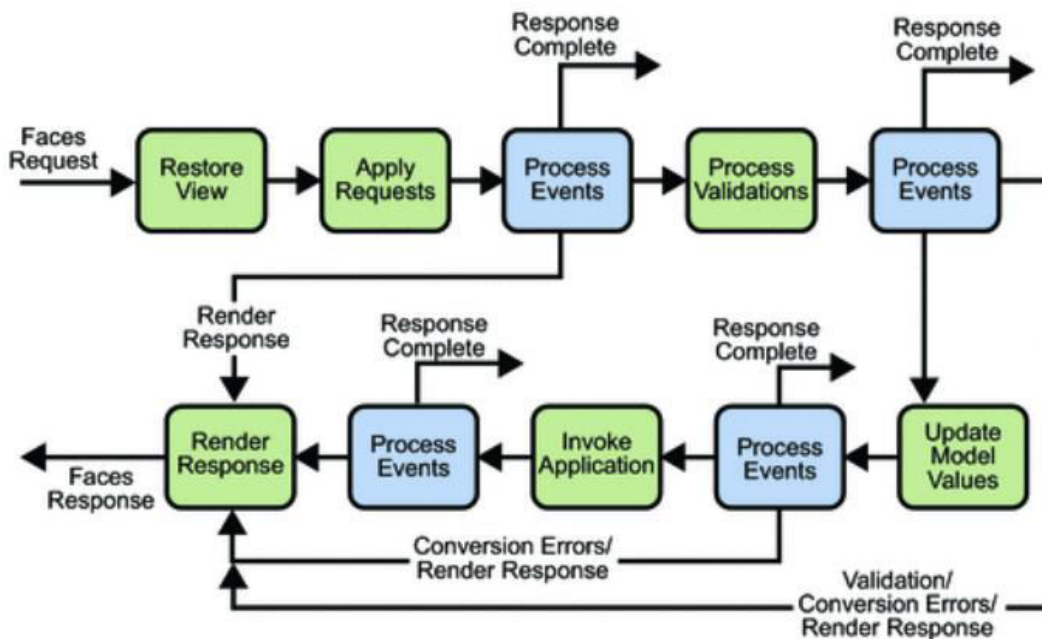
JSP (Java Server Pages)

- textové soubory obsahující:
 - o statická data (většinou HTML či XML)
 - o JSP elementy pro tvorbu dynamického obsahu
- JSP jsou kompilovány do Servletů a v response vráceny klientovi
- JSTL = JSP Standard Tag Library – forEach, if, out,...



JSF (Java Server Faces)

- component-based framework pro server-side UI
- dvě hlavní části:
 - o API reprezentující UI komponenty a řídící jejich stav
 - handle events
 - server-side validace
 - defining page navigation
 - ...
 - o custom JSP tag knihovny pro expressing UI komponent
- díky komponentám lze jednodušeji vytvářet komplexní aplikace
- existuje mnoho knihoven pro komponenty (některé přidávají podporu pro AJAX)



re: JSF lifecycle. From <http://docs.oracle.com/javaee/5/>

Nevýhody JSP/JSF:

- jsou založeny na request/response → časté reloady stránek
- skoro vše se odehrává na serveru → náročné na výkon
- limitovaná podpora AJAX, mobilních zařízení
- obtížné vytváření nových či rozšiřování existujících komponent

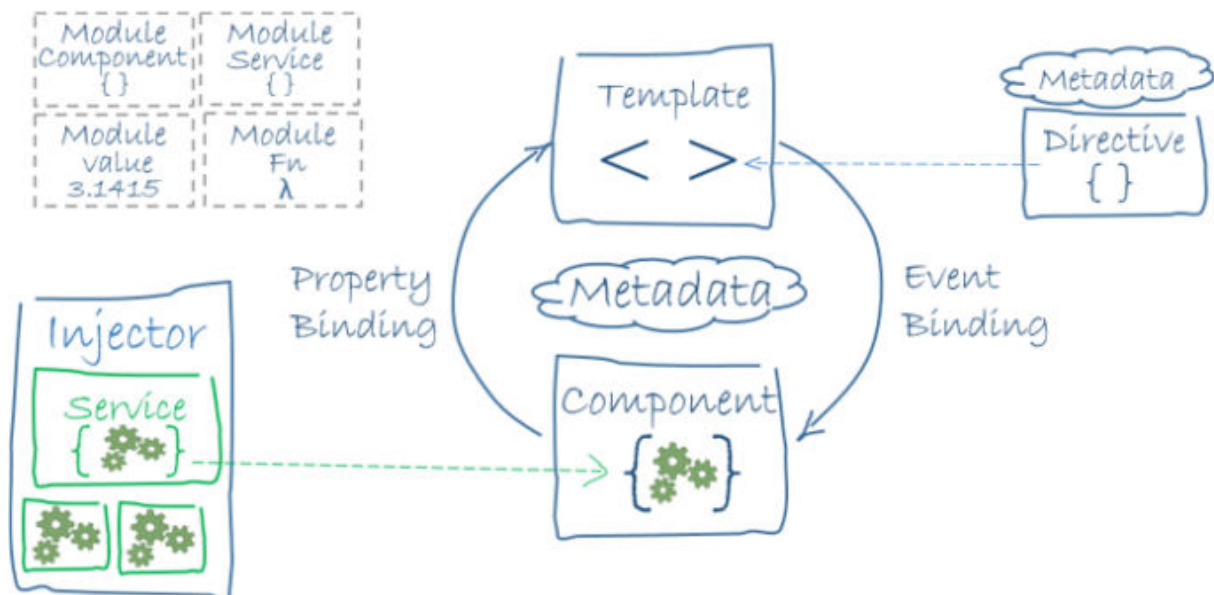
JavaScript-based

- založené na AJAX
 - o asynchronní zpracování
 - o eventy
 - o callbacks, promises
 - o když se něco pokazí, je těžké traceovat stav aplikace
- nedoporučuje se užití jQuery (chaotický kód)
- aplikace většinou odpovídají manipulací DOM stromu dané stránky

- méně refresh/reload
- komunikace se serverem probíhá v pozadí
- většinou single-threaded
- klasifikace:
 - **deklarativní:**
 - „HTML“ šablony s bindováním
 - např. Angular
 - **procedurální:**
 - view-struktura je definována v JS kódu
 - např. React

Angular

- vyvíjen Googlem
- podpora MVC s two-way binding
- HTML šablony rozšířeny o hooks pro JS controllery



React

- vyvíjen Facebookem
- lehce integrovatelný do starších web aplikací
- React Native pro vývoj nativních aplikací pro iOS, Android a UWP

Integrace JS frontendu s backendem

- JS-based frontend komunikuje s backendem pomocí REST web services
- většinou používá JSON jako formát dat
- asynchronní
 - pošle request
 - zpracovává mezitím další věci
 - zavolá callback/resolve Promise když obdrží odpověď

Single Page Applications

- **multi-page aplikace:** individuální stránky používají JS, ale prohlížeč stále naviguje – URL se mění a stránky reloadují [GitHub, FEL GitLab,...]
- **single-page aplikace:**
 - o žádná navigace prohlížeče, vše se odehrává na jedné stránce pomocí manipulace s DOM [Gmail, YouTube,...]
 - o plynulejší user experience
 - o žádné reloady
 - o většina práce na straně klienta
 - o komunikace se serverem na pozadí
 - o důležitá architektura klienta
- skoro vše se musí načíst při otevření stránky (framework, css, application bundle,...)
- jiný způsob navigace a zajištění bezpečnosti
- **nevýhody:**
 - o špatná podpora pro záložkování
 - o navigace, podpora funkce „zpět“
 - o event cancelling
 - o záložkování
 - o SEO
 - o automatizované testování

Klientská architektura

- JS-based klienti se stávají komplexnějšími → nutnost správné struktury

MVC (Model View Controller)

- klasický pattern, aplikovatelný i na client-side JS
- Controller řídí user interaction a navigaci, žádná business logika

MVVM (Model View View-Model)

- View-Model je abstrakcí View
- nechává na frameworku bindování UI komponent na View-Model atributy (two-way binding)

Flux

- jednosměrný
- původ v Reactu
- odděluje business logiku od UI komponent
- umožňuje spravovat globální stav aplikace

