



**Wicket – přehled technologie**  
**Martin Strejc**  
**[www.etnetera.cz](http://www.etnetera.cz)**  
**5.4.2012**



# *Osnova přednášky*

1. Vznik Wicketu
2. Co Wicket umí a co neumí?
3. Účely užití – výhody a nevýhody
4. Rozšiřitelnost Wicketu
5. Srovnání s jinými technologiemi
6. Jak psát aplikace?
7. Malá ukázka Wicketu

## Vývoj

- Jonathan Locke, 2004
- Verze 1.0 uvolněna 2005, začlenění k Apache
- Udržovaná verze 1.4
- Současná verze 1.5 (dost podobná 1.4)
- Chystá se 1.6 s radikálnějšími změnami

# *Přehled technologie*

1. MVC framework
2. Komponentově orientovaný
3. Podobné technologie: JSF, Tapestry
4. Ideologicky blízký Swing (komponenty)

## Co Wicket umí?

1. Generovat HTML, event. XML
2. Generovat formuláře, obsluhovat AJAX (vlastní implementace nebo jQuery)
3. Perzistovat data (stránku, formuláře)
4. Starat se o integritu dat při zpracování formulářů
5. Generovat obrázky, další smíšený binární obsah
6. Konzistentně udržovat URL (vč. čistých URL)

## Co Wicket NEumí?

1. Administrovat obsah (CMS)
2. Používat různé šablony vzhledu
3. On-line měnit obsah view vrstvy či funkčnost  
(ve srovnání s JSP nebo PHP,  
kde lze vyměnit konkrétní soubor)
4. Umístit kód do VIEW (HTML šablony)  
malá výjimka: Apache Velocity

# Výhody Wicketu

1. Relativně rychlé pochopení základních Komponent
2. Oddělenost HTML/markupu a kódu
3. Jednoduchá součinnost grafika (HTML/CSS) a Java programátora
4. Rozsáhlá modifikovatelnost libovolné komponenty nebo chování frameworku
5. Dobrá řetěžitelnost od abstraktního chování k implementacím, přehledné dědění od existujících komponent
6. Validní HTML/XML

## Výhody Wicketu 2

1. Vysoká kompaktnost datového modelu a dobrá udržitelnost dat
2. Jednoduché zpracování a validace formulářů, včetně wizardu (vícekrokový formulář)
3. Jednoduchá implementace AJAXu (built in)
4. Podpora integračních testů (Wicket tester)
5. Knihovny 3. stran
6. Integrace s dalšími frameworky (Spring/MVC)
7. Podpora v IDE (Eclipse - qWickie)



# *Srovnání technologií - vhodnost*

1. Webové aplikace, intranet – Wicket
2. Veřejné weby, portály s vysokou návštěvností  
– CMS, Spring MVC, PHP
3. Agilní vývoj – Ruby on Rails, jiné technologie

# Rychlost a výkon

1. Nic se nekompiluje za běhu (viz JPS, PHP)

2. Více paměti – sporná doměnka

<http://ptrthomas.wordpress.com/2009/09/14/perfbench-update-tapestry-5-and-grails/>

	wicket 1.4.1	tapestry 5.1.0.5	seam 2.2.0.GA	grails 1.1.1
Heap Size (bytes)	9,396,712	22,337,464	24,582,288	63,047,440
Classes	4,478	5,612	6,445	7,919
Objects	158,782	466,136	600,283	1,478,057
Retained Size for 20 HttpSession-s (bytes)	1,346,848	1,215,072	15,333,672	1,363,560

## *Další knihovny*

1. Wicket extensions (přímo s Wicket)
2. Wicketstuff, contrib... (GMAP, jQuery)
3. Integrace se Spring (Spring Core, Spring MVC, Spring Security)
4. Možnost další integrace s obecným servletem
5. Možnost dopsat další vlastní knihovny

... vždy je třeba myslet na udržitelnost kódu

## *Dělám projekt ve Wicketu*

1. Ujistím se, že v týmu mám někoho, kdo již Wicketu rozumí (a nenalzene překážky)
2. Dobře promyslím datový model (DB a Java)
3. Navrhnu vrstvy aplikace, aby byla testovatelná
4. Promyslím integraci se Spring (event. Spring Security, Spring MVC) a dalšími knihovnamí
5. Napíšu aplikaci
6. Napíšu testy!!! (JUnit, Wicket)
7. Fakturuji zákazníkovi :-)

## *Dodržuji následující pravidla!*

1. Vícevrstvá aplikace, čistý kód
2. Recykluji panely a opakující se kód
3. Dávám pozor na křížové závislosti (Wicket, DAO, model, entita, servlet)
4. Píšu testy!
5. Čtu dokumentaci (nepíši znovu již napsané)
6. Knihovny 3. stran používám s rozumem

## *Na co si dát pozor?*

1. Opakující se kód!
2. Opakující se markup (šablona, HTML)
3. Nic není bezchybné, ani Wicket
4. JS a AJAX – pozor na verze prohlížeče
5. Používání návrhových vzorů (singleton)

## Jak to funguje?

1. Wicket je servlet, který poslouchá na určité URL a dodává data
2. Každé view je reprezentováno kódem (HTML/markup), který se načte a sestaví kód výsledný pomocí kontroleru (funguje vkládání – panely, fragmenty)
3. Data jsou reprezentována nejčastěji modelem `IModel<T>`
4. Vpodstatě to vypadá opravdu jednoduše

## Závěr o Wicketu

1. Dobrá technologie, zejména pro aplikace
2. Vyžaduje specifický přístup k vývoji
3. Dává řadu možností pro jednoduchý vývoj kvalitních a testovatelných Java aplikací
4. Při rozhodování o projektu je třeba rozmyslet, zda je Wicket vhodný – posoudit účel



## *Dotazy*

1.... ?

2.... ?

3.... ?

4.... ?

Děkuji za pozornost :-)

[www.etnetera.cz/akce](http://www.etnetera.cz/akce)



# POZDRAV OD EJČ-ÁRA

Ahoj všem,

jsem rád, že jste dneska přišli na náš seminář na téma "**Apache Wicket**„. Nemohl jsem být až do konce a proto budu rád, když mně pošlete zpětnou vazbu – jak se vám to dnes líbilo, co bychom měli do příště zlepšit.

[hr@etnetera.cz](mailto:hr@etnetera.cz) .



# POZDRAV OD EJČ-ÁRA

Zároveň budu moc rád, pokud dorazíte příště, takže neváhejte sledovat náš web:

<http://www.etnetera.cz/cz/akce>

V případě informací o pracovní spolupráci, se podívejte na web: [www.etnetera.cz](http://www.etnetera.cz) nebo prostě napište na [hr@etnetera.cz](mailto:hr@etnetera.cz).

Těšíme se na další shledání  
Ejč-ár a tým Et netera.

