

GoSpot
Semestrální práce
X33EJA
Zimní semestr 2010

Ondřej Mirtes
České vysoké učení technické v Praze
Fakulta elektrotechnická
mirteond@fel.cvut.cz

Václav Purchart
České vysoké učení technické v Praze
Fakulta elektrotechnická
purchva1@fel.cvut.cz

27. ledna 2011

Gowalla je geolokační sociální síť, ve které si uživatelé přes mobilní aplikaci zapisují, jaká místa navštívili a na těchto místech sbírají a vyměňují virtuální předměty. Některých předmětů je přebytek, některé jsou vzácnější a existují i předměty s velice omezeným počtem kusů. Naše webová aplikace si klade za cíl uživatelům Gowally usnadnit lov předmětů – poradí jim, kam do okolí mají vyrazit.

1 Návrh

Uživatel si po přihlášení do naší aplikace bude moci naklikat, jaké předměty chce sledovat. Aplikace pak pomocí scheduleru bude pravidelně kontrolovat místa v okolí uživatelské lokace a v případě, že se tam vyskytne kýžený předmět, uživateli pošle e-mail. Frontend aplikace je tedy velice jednoduchý, více práce se bude odehrávat na pozadí v modelu.

2 Realizace

Pro implementaci jsme použili několik knihoven třetích stran. Nejdůležitější knihovnou je GowallaJava¹, která poskytuje rozhraní pro komunikaci s Gowalla API. Komunikace s ním probíhá ve formátu JSON a odpovídá RESTful² specifikaci.

Tato knihovna zajišťuje i autorizaci uživatele přes OAuth³. Ta probíhá tak, že uživatele z naší aplikace nasměrujeme na speciální URL na webu Gowally, kde potvrdí, že naší aplikaci chce povolit přístup k jeho datům. Po potvrzení Gowalla přeměruje uživatele zpět k nám a na základě parametru v URL, který si uložíme do naší databáze, můžeme začít komunikovat s Gowalla API.

Model naší aplikace je realizován pomocí anotovaných JPA⁴ entit. Fasády pak zaobalují práci s entity managerem a poskytují další metody jako prostředník mezi naší databází a Gowalla API, např. pro aktualizaci vyskytujících se předmětů či aktualizaci aktuální uživatelské lokace (na základě jeho nejnovějšího check-inu).

Anotace `@Schedule` nad kýženými metodami pak v GlassFish⁵ serveru zařizuje pravidelné spouštění. Pro tyto účely máme třídu `CronFacade`, která aktualizuje uživatelskou lokaci a vyskytující se předměty v okolních místech a `SendReportFacade`, která uživatelům odesílá e-maily s reportem.

Pro uživatelskou aplikaci jsme použili JSF⁶ společně s nadstavbou PrettyFaces⁷, která zpřístupňuje práci s frameworkem např. podporou hezkých URL nebo možností spouštět jakýkoli kód v Javě pouze na základě přístupu na URL (potřebovali jsme pro OAuth autorizaci), což JSF v základu neumí.

¹<http://code.google.com/p/gowalla-java/>

²http://en.wikipedia.org/wiki/Representational_State_Transfer

³<http://en.wikipedia.org/wiki/OAuth>

⁴http://en.wikipedia.org/wiki/Java_Persistence_API

⁵<http://glassfish.java.net/>

⁶http://en.wikipedia.org/wiki/JavaServer_Faces

⁷<http://ocpsoft.com/prettyfaces/>

3 Problémy

Největší problémy jsme zaznamenali při komunikaci s Gowalla API. Jednak není řádně zdokumentováno, takže některé parametry jsme museli hledat v jiných zdrojích na webu a jednak je nestabilní. Často nám vrací kódy 403 (Forbidden) nebo 500 (Server error).

Stejně tak autorizace s OAuth nefunguje přesně dle specifikace a nedaří se nám obnovit přístupový token (který do 14 dnů od získání vyprší a je třeba získat nový).

4 Závěr

Pokud bychom aplikaci chtěli reálně provozovat, bylo by třeba na ni udělat ještě hodně práce, zejména zefektivnit a ošetřit práci s Gowalla API, které často nefunguje. Dále by bylo třeba implementovat chytřejší zasílání e-mailů, které by si zaznamenalo, na jaké předměty jsme již upozorňovali, abychom uživatele nebombardovali stejnými e-maily několikrát za sebou.

Nicméně si myslíme, že jsme vypracovali originální a zajímavou semestrální práci, která využívá nevšední nástroje (OAuth, EJB Scheduler, PrettyFaces).