



Introductie

Voor deze case maken we gebruik van een veel voorkomende situatie: het importeren van gegevens van derde partijen. Deze derde partijen bieden in allerlei vormen data aan die worden gebruikt voor het samenstellen van het aanbod van reizen op sites van klanten.

Er moet een applicatie gebouwd worden waarmee twee csv bestanden geïmporteerd en gekoppeld worden, om vervolgens bepaalde hotels op te vragen. Zie Opdracht voor een gedetailleerde omschrijving van de te ontwikkelen applicatie.

Omgeving

Werk met minimaal Python 2.6 en Django 1.3.

Data

De data om mee te werken betreft twee CSV bestanden, te vinden via een afgeschermd URL (HTTP Basic Authentication):

<http://rachel.maykinmedia.nl/djangocase/city.csv>

<http://rachel.maykinmedia.nl/djangocase/hotel.csv>

Username : python-demo

Password : claw30_bumps

De bestanden zijn ook attached bij deze e-mail, mocht “authenticated HTTP” niet lukken dan kun je deze bestanden gebruiken voor de import.

Opdracht

Hieronder staan een aantal punten waaraan de opdracht moet voldoen.

Hoofd punten

- Zorg ervoor dat alle data middels Django modellen in de database komt te staan en “slim” aan elkaar gekoppeld wordt. (Een hotel ligt in een bepaalde stad)
- De logica om de data te importeren via de URLs moet aanwezig zijn. Een eenmalige import is niet de bedoeling. Mocht “authenticated HTTP” niet lukken, dan met een file import.
- Maak een view/template/form die gebruik maakt van de geïmporteerde data om:
 - Een stad te kiezen (steden welke voorkomen in de data)
 - Bij de gekozen stad alle hotels uit die stad weer te geven
- Zorg ervoor dat je middels unit tests de kwaliteit van je werk kan aantonen, wees hier pragmatisch in.
- Documenteer je code zodanig dat een andere developer verder zou kunnen gaan waar jij bent gebleven. Ook hier geldt: wees pragmatisch.

Optionele punten

- Maak een eenvoudige admin interface in Django om de data in te zien en te bewerken.
- Gebruik JavaScript om bij het kiezen van een stad automatisch de hotels op te vragen, eventueel met auto-complete functie die gebruik maakt van AJAX.
- Schrijf de import functionaliteit als een Django Command, zodat deze in een cron-job kan worden opgenomen.
- Maak gebruik van een repository systeem als svn, hg, git, etc . In dit geval willen we graag naast het eindresultaat, ook de tussentijdse wijzigingen in kunnen zien.

Tot slot

Mocht iets niet lukken, ga dan verder met het volgende onderdeel of pak wat optionele punten op. Documenteer in dat geval beknopt in de code wat je wilt doen. Je bent verder vrij om van elke applicatie, tool, bron of voorbeeld gebruik te maken.

Als iets niet helemaal duidelijk dan kan je altijd mailen.

Succes,

Joeri & Alex