



WERKGROEP GAP		202
21022012	Not.	2

Aanwezig: Bart Boone, Sven Maes, Tom Haepers, Johan Vervloet
Verontschuldigd: Koen Meersman, Mattias Deparcq, Broes De Cat, Tim Mallezie

Verder na vorige vergadering

Er zijn tijdens de vorige vergadering een hoop feature requests verzameld. Waar gaan we nu aan werken:

Useability

Veel vragen gaan over gebruiksvriendelijkheid. Sven wil werken aan een nieuwe user interface. Dit zijn de eerstvolgende stappen:

- Johan lijst de use cases op
 - Features die we al hebben (overlopen van de controller actions)
 - Features die we nog willen implementeren
- Sven werkt de use cases stap voor stap uit, en tekent wireframes
- De wireframes moeten dan nog geïmplementeerd worden.

Het meeste programmeerwerk hiervoor zal waarschijnlijk in de frontend moeten gebeuren. Maar er zullen waarschijnlijk ook wel nieuwe service methods of aanpassingen van bestaande service methods nodig zijn.

(Sven, van zodra je een aantal wireframes hebt, bezorg ze ons, zodat we er feedback op kunnen geven. Tim is trouwens ook geïnteresseerd in useability, misschien kunnen jullie wel samen werken.)

Gebruikersrechten

Er is vraag naar gebruikersrechten. Hoewel we dat nog niet direct gaan implementeren, moeten we met deze toekomstige feature rekening houden bij het opstellen van de wireframes. We denken aan deze gebruikersrechten:

- Iemand die alle gegevens van zijn groep kan aanpassen (onze huidige GAV)
- Iemand die zijn eigen (persoonlijke) gegevens kan aanpassen, en de gegevens van de leden van zijn afdeling
- Iemand die enkel kan lezen

(leesrechten heb je altijd op alles van je groep)

API

Er is al een API-vergadering geweest met Ben Bridts, en er staat al wat documentatie op de wiki.

<https://develop.chiro.be/trac/cg2/wiki/API>

In verband met de **implementatie** van OData met WCF Data Services. IQueryable zelf implementeren zou een erg interessant maar erg moeilijk project zijn. Als Ben Bridts of Mathias Keustermans dat wil doen, dan mag dat uiteraard. Maar omdat het telkens om een beperkt aantal personen per groep gaat, en omdat het aantal gebruikers van de API ook beperkt zal zijn, mag er net zo goed een van de alternatieve werkwijzes gebruikt worden:

- De API vraagt telkens alle leden van de groep op (eventueel cachen), en we laten Data Services de rest van het werk doen
- We kunnen ook data services voorzien voor specifieke operaties (bijv. ophalen van details van 1 persoon), zie <http://msdn.microsoft.com/en-us/library/cc668788.aspx>

(een bedenking van het moment: Als Ben of Mathias IQueryable niet implementeren, dan is de implementatie van IQueryable voor GAP misschien wel een interessant project voor een student)

In verband met de **authenticatie**: in het verslag van de API-vergadering (<http://ur1.ca/8atqc>) staat dat bij basic HTTP-authenticatie het gevaar bestaat dat username en wachtwoord gewoon via de javascripts bij de user terecht komt. Dat is een terechte bezorgdheid, maar dat geldt even goed voor bijvoorbeeld OAuth. We passen onze strategie dus aan:

- We gaan JSONP opnieuw disablen, zodanig dat de API niet meer rechtstreeks aangeroepen kan worden vanuit een javascript. Aanroep van de API moet vanop de server gebeuren (bijv. in een php-script).
- We gebruiken gewone basic authenticatie (over https weliswaar) om aan te melden op de API.

- JQuery kan nu gebruikt worden via een omweg. Je maakt een server side php-script dat gewoon het resultaat van de service call echo't (een proxy als het ware). JQuery werkt dan op de resultaten van die proxy.

Over de accounts:

- We gebruiken het huidige usersysteem voor gebruikers van de API. Een API-account krijgt dus ook een vervaldatum, en is verlengbaar.
- Hoogstens 1 groep per API-account. Op die manier weten we na authenticatie direct uit welke groep we de gegevens mogen ophalen.
- We moeten de API-gebruikers als dusdanig kunnen herkennen, om het volgende af te dwingen
 - API-users krijgen geen toegang tot de gewone webinterface
 - met gewone user credentials kun je niet aanloggen voor de API

(Ben, Mathias: contacteer me gerust als jullie vragen hebben)

Nightly builds

Koen wou wel eens bekijken of hij automatische builds kon opzetten, d.m.v. powershellscripts. Johan deed hiervoor al wat voorbereidend werk op devsrv1, en maakte een nieuwe dev- en testdatabase in sql server 2008 r2.

Tommy raadt aan om eens te kijken naar [cruisecontrol.net](http://www.cruisecontrolnet.org/projects/ccnet/wiki), waarvan de url verrassend de volgende is:

<http://www.cruisecontrolnet.org/projects/ccnet/wiki>.

Deployment van de automatische build kan in eerste instantie gewoon door de juiste files te kopiëren. Op termijn kunnen we ook met MSI-pakketten werken.

Development workflow, git

Een aantal weken terug werd een voorstel gemaild in verband met een git-setup waarbij een 'dictator' (johan ;) de code kan nakijken vooraleer ze in de 'upstream' terecht komt. We hebben infrastructuur voor git. Johan zet iets op, waarmee de geïnteresseerden dan kunnen experimenteren, om te zien hoe het loopt.

Het is alleszins mogelijk om een subversion-repository als een remote repository te beschouwen in een git-setup. Maar hiermee moet nog wat geëxperimenteerd worden. Ook moet onderzocht worden of git op een elegante manier in Visual Studio geïntegreerd kan worden.

Hoe dan ook is het OS van develop.chiro.be (Debian Lenny) net out of support gegaan. Een upgrade van het OS van die server naar Debian Squeeze moet sowieso gebeuren, omdat er nog wel wat projecten daar in de trac/svn zitten.

Weekend

We gaan op gezamenlijk programmeerweekend met de internetploeg: GAPIP 2. De datum ligt vast: 8 – 10 juni.

Varia

- Er zijn nieuwe scripts om de gegevens voor de database te anonimiseren. Op de DEV-database hebben die al gelopen. Op de nieuwe test-database ook (denk ik), maar de publieke test gebruikt nog de oude testdatabase. Ik wou dat rap fixen, maar blijkbaar heeft de service account voor de test-backend nog geen toegang tot die nieuwe database. Dit wordt zo snel mogelijk rechtgezet door Johan.
- Johans 'vrijwilligerspolicy': als je (als vrijwilliger) iets aan GAP wilt implementeren omdat je dat interessant vindt, dan mag je dat altijd doen.
 - Kleine changes en fixes mag je committen in de trunk (maak wel een ticket aan als dat nog niet moest bestaan)
 - Voor fundamentele zaken moet je een branch maken
 - Zit je met een vraag of probleem: aarzel niet om Johan te contacteren.
 - De architectuur 'backend', 'services', 'frontend' moet behouden blijven. Voor de rest mag je doen wat je wilt.
 - Als het resultaat goed is, wordt het gedeployd. (Mogelijke discussies over goed of niet goed worden op de werkgroepvergadering of via de maillijst gevoerd.)
- De wireless router op de IT-bureau zou nog eens geconfigureerd moeten worden als bridge (ipv nu met natting en een eigen subnet). Op die manier kun je SQL Server Management studio ook via de wireless connectie gebruiken.
- Application data redirection is hinderlijk voor vrijwilligers die op devsrv1 moeten developen