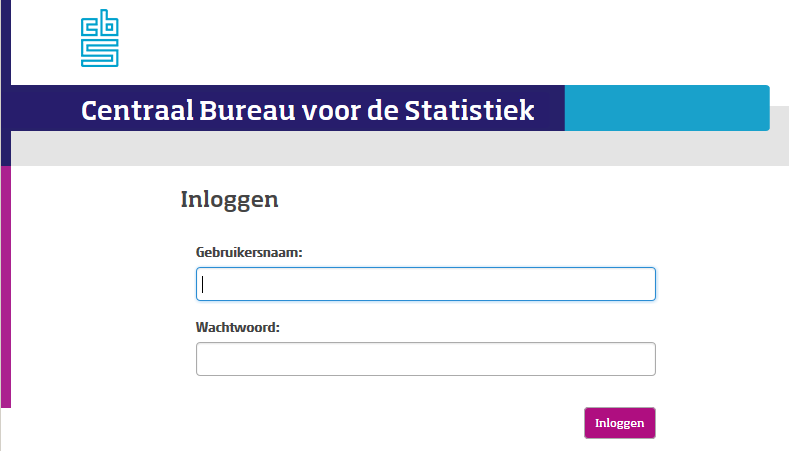
# Requirements ClockBot

In opdracht van Phoenix Transitie moet een clockbot ontwikkeld worden die geautomatiseerd de laadtijd van ‘pageturns’ moet meten in Blaise vragenlijsten.

Locatie vragenlijsten

Alle te testen vragenlijsten worden door DVZ altijd op dezelfde website <http://prepenquete.cbs.nl> aangeboden. Door de juiste combinatie van *gebruikersnaam* en *wachtwoord* te gebruiken wordt de juiste vragenlijst geopend. Deze worden door DVZ veelal in een xlsx tabelletje aangeleverd. De clockbot zou dit moeten kunnen uitlezen. [Hier kunnen we een vast format voor afspreken.]



Browsers

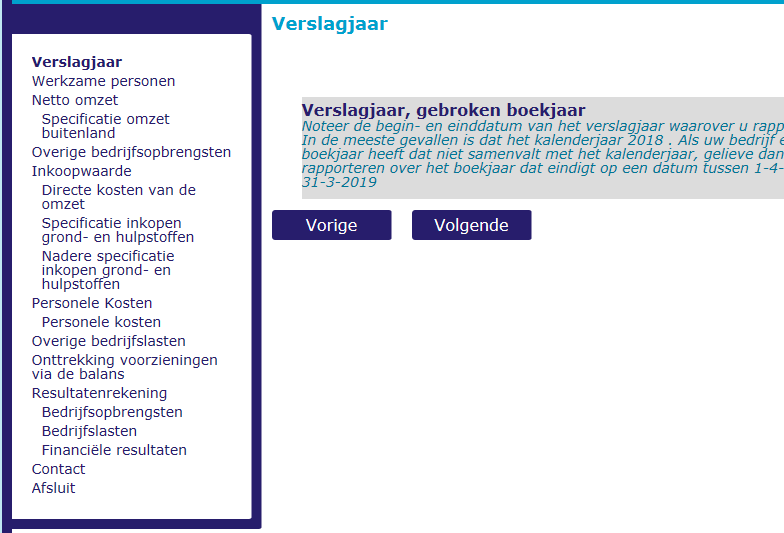
De laadtijd blijkt nogal afhankelijk te zijn van de browser die gebruikt wordt. In ieder geval de volgende browsers moeten kunnen worden getest:

* Internet Explorer (eventueel via de citrix login van het CBS?)
* Edge
* Chrome
* Firefox
* Safari

Wellicht is het handig wanneer er een mogelijkheid komt te selecteren welke browser je wilt testen (of juist niet wil testen).

Laadtijd pageturn meten

Wanneer je ingelogd bent krijg je de volgende pagina-opbouw te zien. Zoals duidelijk wordt kan de laadtijd van de pageturns nu op twee manier gemeten worden, met de index-methode of met de volgende-methode.



Index-methode

Alle nieuwe vragenlijsten die gemaakt worden voor Phoenix krijgen een index (het linker deel). Meten met de indexmethode betekent dan het volgende:

1. Na de inlogprocedure sta je op het eerste indexitem (als het goed is)
2. Klick op het volgende indexitem (hier ‘werkzame personen’)
3. Meet de laadtijd van de pageturn
4. Klick op het eerste indexitem (‘verslagjaar’)   
   De laadtijd blijkt beïnvloed te worden door zowel de pagina die je verlaat als de pagina waar je naartoe gaat; daarom willen we voor elke meting dezelfde “startpagina” gebruiken.
5. Klick op het volgende indexitem (‘netto omzet’)
6. Meet de laadtijd van de pageturn
7. Is het indexitem de laatste (hier ‘Afsluit’)?
   1. Nee, ga verder met 4.
   2. Ja, ga verder met 8
8. Klick op het eerste indexitem (‘verslagjaar’)
9. Meet de laadtijd van de pageturn   
   De laadtijd van het eerste indexitem willen we ook nog meten zonder dat daar de logintijd bij zit.
10. Klaar

Volgende-methode

Naast de indexmethode kan ook de Volgende-methode gebruikt wordt. Onder elke pagina staat altijd een knop ‘volgende’ waarmee door de hele vragenlijst gebladerd kan worden. Deze methode zou er dan als volgt uit zien:

1. Na de inlogprocedure sta je op het eerste indexitem (als het goed is)
2. Klick op de knop volgende
3. Meet de laadtijd van de pageturn
4. Is het indexitem de laatste (hier ‘Afsluit’)?
   1. Nee, ga verder met 2.
   2. Ja, ga verder met 5
5. Klaar

De meting in grote lijnen

Het protocol voor de meting als geheel gaat er als volgt uit zien:

1. Selecteer de eerste browsers
2. Selecteer de gegevens van de eerste gebruiker
3. Open *website* in *browser*
4. Log in met *gebruikersnaam* en *wachtwoord*
5. Meet direct de laadtijd van de inlogprocedure
6. Meet de laadtijd via *methode x*
7. Sluit de browser   
   elke meting moet gestart worden vanuit een fris geladen browser
8. Was dit de laatste gebruiker?
   1. Nee, ga naar 9
   2. Ja, ga naar 10
9. Selecteer de gegevens van de volgende gebruiker en ga naar 3
10. Was dit de laatste browser?
    1. Nee, ga naar 11
    2. Ja, ga naar 12
11. Selecteer de volgende browser en ga naar 2
12. Stel een resultaten bestand op

Overige punten

🡪 Tijdens het meten moet bijgehouden kunnen worden van welke pagina nu precies de laadtijd gemeten wordt. Dit moet namelijk meegegeven worden in de resultaten. Deze moet zowel met de index- als met de volgende-methode bijgehouden kunnen worden.

🡪 Misschien kan nog ingebouwd worden dat dezelfde gebruiker in dezelfde browser meerdere keren achter elkaar gemeten wordt? Om zodoende een gemiddelde laadtijd te meten.

🡪 We moeten nog even overleggen hoe het resultaten bestand er precies uit komt te zien. Ik zou zeggen dat in ieder geval de volgende variabelen erin opgenomen moeten worden:

Browser

Gebruikersnaam

Paginanaam

Laadtijd