BPC eksempelklient versjon 1.1 27.10.2009

Programmert av Audun Nystad Bugge, audunnys@stud.ntnu.no.
Ta kontakt ved spørsmål eller problemer.

Filer i pakken:

- readme.pdf - denne fila, dokumentasjon
- readme.txt – samme som readme.pdf, men i rent tekstformat
- changes.txt - changelog, brukes som oppdateringsguide for systemer som allerede har implementert en tidligere versjon av denne klienten.
- bpc_core.php - inneholder funksjonen bpcDoRequest() som tar seg av den mest grunnleggende tilkoblingen til BPC. Du kan enten bruke denne som den er, eller implementere den som du vil i egen kode.
- config.php - innstillinger brukt av bpc_core.php.
- sample.php - er et eksempel på hvordan en klient for BPC som bruker bpc_core.php kan lages.

Denne dokumentasjonen med eksempler antar at dere bruker en web-server med PHP versjon 5 eller nyere med støtte for cURL installert.

Konfigurasjon

I config.php ligger noen få innstillinger. Her vil dere endre id og key for deres linjeforening, og sannsynligvis ikke noe annet

Argumenter og funksjoner

Alle argumenter BPC trenger for å utføre forespørselen må inkluderes i et array, her brukes $postdata. Parametere som alltid må være med er som følger:

- **forening** - linjeforeningens id i systemet.
- **key** - linjeforeningens unike kode.
- **method** - hvilket format dataen som returneres skal ha. Foreløpig er kun 'serialized_array' støttet, det vil si et php-array som er serialized og kan brukes direkte i php-koden etter å ha kjørt unserialize(). Meld fra på audunnys@stud.ntnu.no hvis andre formater ønskes, så ordner jeg det.
- **version** - hvilken versjon av bpc-klient som brukes. Dette settes i bpc_core av oss. Hvis dere lager deres egne funksjoner setter dere version til det samme som den klientversjonen dere er oppdatert i forhold til, dette for å sikre at riktig data kreves i input og returneres tilbake.
- **request** - hva serveren skal utføre, se neste avsnitt

De tre første av disse, forening, key og method, blir satt automatisk fra verdiene i config.php. Hvis det er ønskelig kan method i tillegg settes direkte for hver enkelt forespørsel via postdata hvis engang flere verdier blir tilgjengelige.

Nå følger en oversikt over støttede verdier av 'request', hvilke tilleggsargumenter som kreves og hvilke som er frivillige samt hvilke data som blir returnert:
Det kan også være til hjelp å bruke dette eksempelskriptet for å se hvilke verdier de forskjellige funksjonene returnerer, da alle mottatte data skrives ut med var_dump().

get_events - henter alle eller et enkelt arrangement

Frivillige argumenter:

- **event** – arrangement-id
- **username** – NTNU-brukernavn
- **fromdate** - Hent kun hendelser etter dette tidspunktet. Format: YYYY-MM-DD HH:MM:SS. Hvis dette ikke er angitt hentes alle arrangementer i framtiden, evt alle før todate
- **todate** - Hent kun hendelser før dette tidspunktet. Format: YYYY-MM-DD HH:MM:SS. Hvis dette ikke er angitt hentes alle arrangementer i framtiden, evt alle etter fromdate
- **event_type –** Kan være enten 'advertised', 'not_advertised' eller 'all'. 'advertised' gir kun hendelser hvor linjeforeningen har inngått avtale om blæstedeal. Dette er standard, og er slikt ting fungerte før versjon 1.1. 'not_advertised' gir da hendelser uten blæstedeal, og 'all' gir begge deler.

Returnert data:
$data['event'][int], int itererer over alle arrangementer

- **id** - Arrangementets id (det du vil bruke i som event-parameter)
- **title** - Navnet på firmaet som holder bedpres
- **description** - beskrivelse av arrangementet
- **time** - Når arrangementet skjer, i formatet YYYY-MM-DD HH:MM:SS
- **place** - Hvor det skjer
- **url** - Link til bedriftens nettside
- **logo** - url til bedriftens logo
- **deadline** - siste frist for påmelding, i formatet YYYY-MM-DD HH:MM:SS
- **deadline_passed –** Om fristen for påmelding er passert, 1 eller 0 (true/false)
- **is_advertised** – 1 hvis dere har blæstedeal, 0 ellers. Trenger bare ta hensyn til denne når event_type='all'
- **registration_start –** Påmelding starter, i formatet YYYY-MM-DD HH:MM:SS
- **registration_started –** Hvorvidt påmeldingen har begynt, 1 eller 0 (true/false)
- **seats** - Antall plasser tilgjengelig for din linjeforening totalt
- **seats_available** - Antall ledige plasser for din linjeforening
- **this_attending** - Antall påmeldte fra din linjeforening. Ikke nødvendigvis det samme som seats-seats_available siden seats_available tar hensyn til plasser reservert for andre linjeforeninger.
- **open_for** - Arrangementet er åpent fra og med dette årstrinnet
- **waitlist_enabled** - 1 hvis venteliste er aktivert for dette arrangementet
- **count_waiting** - Antall personer på venteliste fra denne linjeforeningen
- **web_page** - url til bedriftens nettsider
- **is_waiting** - Kun hvis username-parameteren er gitt, angir om denne brukeren står på venteliste (1) eller ikke (0).
- **attending** - Kun hvis username-parameteren er gitt, angir om denne brukeren er påmeldt (1) eller ikke (0). Hvis personen står på venteliste gir attending 0.

add_attending - Melder en person på et arrangement

Påkrevde argument:

- **fullname** - Fullt navn
- **username** - NTNU-brukernavn

- **card_no** - SHA1-hash av kortnummeret. Det er viktig å konvertere kortnummeret til en float eller på annen måte fjerne 0-ene først i kortnummeret før det hashes.
- **event** - arrangement-id (i BPC, som gitt av get_events)
- **year** - årstrinn i studiet

Frivillig argument:

- **user_id** - brukerens id i deres database. Kan være hendig å ha for enklere koding med get_attending, men kreves ikke av oss.

Returnert data:
['add_attending'][0] = 1 ved vanlig påmelding uten feil. Hvis brukeren blir satt på venteliste er ['add_attending'][0] et array med 'waiting' som key og brukerens plass på ventelista som value: array('waiting' => int). Hvis en feil oppstod kommer det en feilmelding i $data['error'].

rem_attending - Melder en person av et arrangement

Påkrevde argument:

- **event** - arrangement-id (i BPC, som gitt av get_events)
- **username** - NTNU-brukernavn

Returnert data:
$data['rem_attending'][0] = 1 hvis det gikk bra, hvis ikke kommer en feilmelding i $data['error']

get_attending - Returnerer alle deltagere fra din linjeforening til et arrangement

Påkrevd argument:

- **event** - arrangement-id (i BPC, som gitt av get_events)

Frivillig argument:

- **sort** - hvordan deltagerne skal sorteres. Alternativer: 'lname' (etternavn, standard), 'fname' (fornavn), 'username', 'registered' (når påmeldingen skjedde)

Returnert data:
$data['users'][int], int itererer over alle påmelde brukere sortert på etternavn

- **user_id** - brukerens id i deres database, tilgjengelig hvis dere tok med denne parameteren i add_attending
- **fullname** - Fullt navn
- **username** - NTNU-brukernavn
- **registered** - når påmeldingen skjedde, i formatet YYYY-MM-DD HH:MM:SS
- **year** - hvilken årstrinn i studiet personen er på

get_user_stats – Returnerer oppmøtestatistikk for et gitt brukernavn

Påkrevd argument:

- **username –** NTNU-brukernavn

Frivillig argument:

- **detailed_stats** – Definerer hva slags data som returneres, se nedenfor.
- **fromdate** – Dato vi skal begynne å regne statistikk fra, format YYYY-MM-DD HH:MM:SS. Standard er at vi regner statistikk fra romerrikets fall til dagens dato.
- **todate** – Dato vi regner statistikk fram til, format YYYY-MM-DD HH:MM:SS
- **event_type –** Kan være enten 'advertised', 'not_advertised' eller 'all'. 'advertised' gir kun hendelser hvor linjeforeningen har inngått avtale om blæstedeal. Dette er standard, og er slikt ting fungerte før versjon 1.1. 'not_advertised' gir da hendelser uten blæstedeal, og 'all' gir begge deler.

Returnert data:
Hvis detailed_stats = true:

$data['event'][int], int itererer over alle arrangementer hvor brukeren var påmeldt eller på venteliste

- **id** - Arrangementets id
- **title** - Navnet på firmaet som holder bedpres
- **time** - Når arrangementet ble avholdt, i formatet YYYY-MM-DD HH:MM:SS
- **is_advertised** – Om linjeforeningen hadde blæstedeal på denne bedpresen
- **attended** – Om brukeren møtte opp (1) eller ikke (0)
- **on_waitlist** – Hvis brukeren ikke møtte opp: Om han var på venteliste (1) eller ikke (0). Hvis han møtte opp er on_waitlist alltid 0, uavhengig av om han stod på venteliste før arrangementet.

Hvis detailed_stats= false:
$data['user_stats'][0]

- **events** – Hvor mange arrangementer brukeren har vært påmeldt eller på venteliste til
- **attended** – Hvor mange arrangementer brukeren har møtt opp på
- **on_waitlist** – Hvor mange av arrangementene brukeren ikke møtte opp på hvor han stod på venteliste

For å finne ut hvor mange «prikker» en bruker har, tar man altså events – attended – waitlist.

get_waiting

Samme parametere som get_attending, men gir personer på venteliste. Sortert etter påmeldingtidspunkt som standard, kan overstyres med 'sort'.

Feilmeldinger

Hvis det skjer en feil, vil det bli returnert som et flerdimensjonalt array i $data['error']. En feil kan være alt fra mangel på kontakt med SQL-serveren til at arrangementet er fullt, så det er viktig å håndtere disse på en skikkelig måte. Koden i bpcRequest() burde gi en ide om hvordan errorhåndtering kan angripes.

Her følger en oversikt over feilkoder sortert etter alvorlighetsgrad.

Fatalt

Disse avbryter skriptet.

101 - Feil eller ingen handshake
102 - Feil eller ingen request
103 - Ingen eller ikke-eksisterende event (get_attending, add_attending, rem_attending)
104 - Feil i SQL-spørring
105 - Feil eller ikke noe brukernavn gitt (add_attending,rem_attending)
106 - Feil eller ikke noe fullstendig navn gitt (add_attending)
107 - Feil eller ikke noe kortnummer gitt (add_attending)

Feil

Alvorlige feil som ikke bør skje med en korrekt oppsatt klient, men ikke så alvorlige at skriptet avbrytes. En fatal feil vil ofte følge etter en vanlig feil, disse kan da hjelpe til å spore problemet.

201 - Feil i validering av en parameter
202 - Du har ikke tilgang til å gjøre dette
203 - PHP-feil i scriptet på serveren

Tilbakemeldinger

Dette er vanlige tilbakemeldinger som kan komme og som bør tas høyde for ved bruk av funksjonene gitt i parentes.

401 - Studenten går ikke på høyt nok årstrinn for dette arrangementet (add_attending)
402 - Det finnes ikke ledige plasser på dette arrangementet (add_attending)
403 - Det finnes ingen arrangementer som passer med denne forespørselen (get_events)
404 - Det finnes ingen deltagere som passer med denne forespørselen (get_attending, get_waiting)
405 - Personen (kortnummer eller brukernavn) er allerede påmeldt dette arrangementet (add_attending)
406 - Brukernavnet er ikke påmeldt dette arrangementet og kan dermed ikke meldes av (rem_attending)
407 - Brukeren var ikke påmeldt som medlem av denne linjeforeningen, og kan dermed ikke meldes av (rem_attending)
408 – Påmelding har ikke startet ennå (add_attending)
409 – Påmeldingen er avsluttet (add_attending)

Tegnsett

All data som returneres fra BPC er med UTF-8-tegnsett. Hvis siden deres f.eks. bruker ISO-8859-1 må dere kjøre utf8_decode($variabelnavn) på alle tekststrenger som kan tenkes å inneholde æ, ø, å og andre ikke-ASCII tegn som mottas fra BPC. På samme måte er det viktig at dere kjører utf8_encode($variabelnavn) på tekststrenger som sendes til BPC, det gjelder da særlig fullname siden man her er garantert å støte borti problemer ellers.