

Notifive Calendar Sync

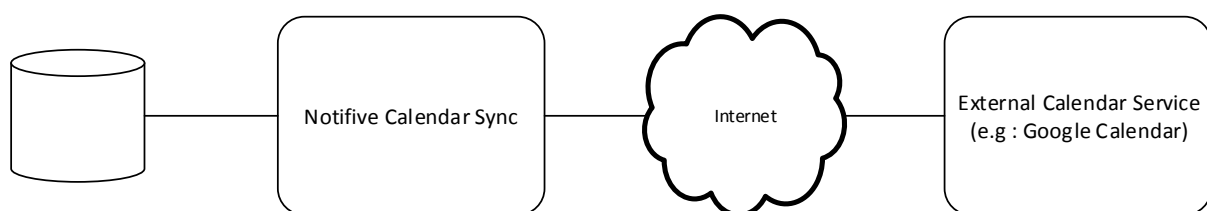
Revision Info

V0.1	General Info, Technical Requirements and Milestone 1	March 11, 2014
V0.2	Additional requirements for 0.2	March 28, 2014
V0.3	Small change in the iCalendar API	April 4, 2014

1 Introduction

The purpose of this software program are the following

1. Develop a web application called “Notifive Calendar Sync” using Java, hence forth called as NCS
2. NCS should read appointment data from its database table and populate the appointment data on an external calendar service like Google Calendar
3. NCS should read appointment data from an external calendar service like Google Calendar and NCS database Application



2 Technical Specifications

1. NCS should use Java 6 or 7
2. NCS should support Tomcat as the application server
3. NCS should support Postgres 9.x with JDBC connection
4. NCS should support OAuth 2.0 for authentication with external servers
5. NCS should NOT store any user name or passwords in databases or in source code
6. All configuration parameters should be configurable via database or web.xml file
7. NCS should support log4j for logging

3 Calendar Sync Web page

Mock up of how the Calendar Sync page should look like is provided below.

Calendar Sync Settings

Select External Calendar(s)

Google	<input checked="" type="checkbox"/>	
Apple	<input checked="" type="checkbox"/>	
Office 365	<input checked="" type="checkbox"/>	

Sync Direction

Both Directions	<input checked="" type="checkbox"/>	
Notifive to External Calendar	<input checked="" type="checkbox"/>	
External Calendar to Notifive	<input checked="" type="checkbox"/>	

Sync Schedule

Auto	<input checked="" type="checkbox"/>	<div style="background-color: #4a86e8; color: white; padding: 5px 10px; border: 1px solid #000; display: inline-block;">Sync Now</div>	<div style="background-color: #00ff00; color: black; padding: 5px 10px; border: 1px solid #000; display: inline-block;">Save Settings</div>
------	-------------------------------------	--	---

NOTE: You will be provided a sample page and CSS to use

NOTE: This page will be repeated for each Notifive Calendar, selectable by dropdown menu or tab

4 How it works

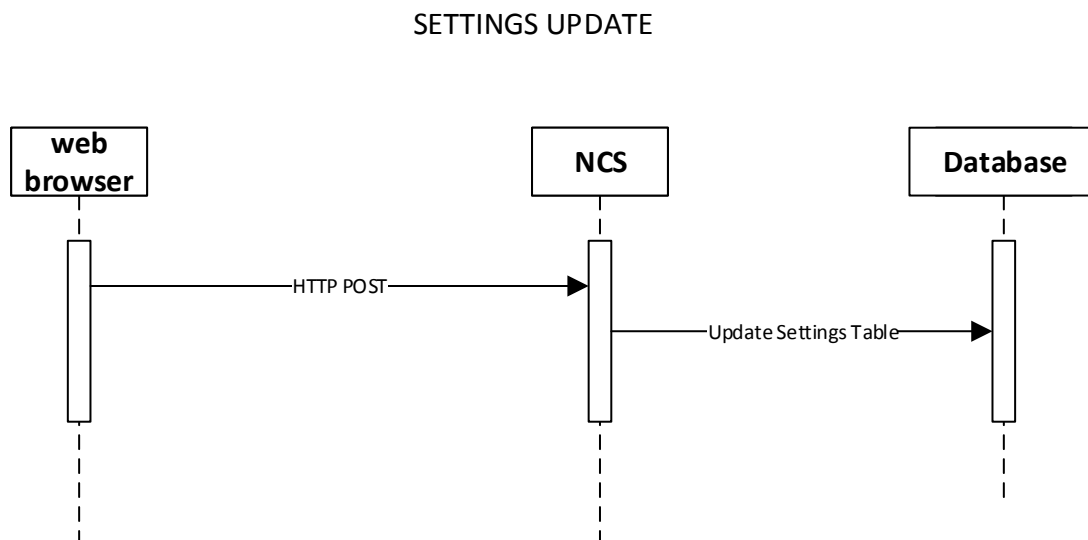
Notifive appointment data will be inserted into the appointment table whenever the user set up an appointment. Depending on the settings, the appointment data will be 'pushed' to the external server.

Appointment data will be pushed to the external calendar either when the data is inserted into the database or when the user clicks the 'Sync Now' button

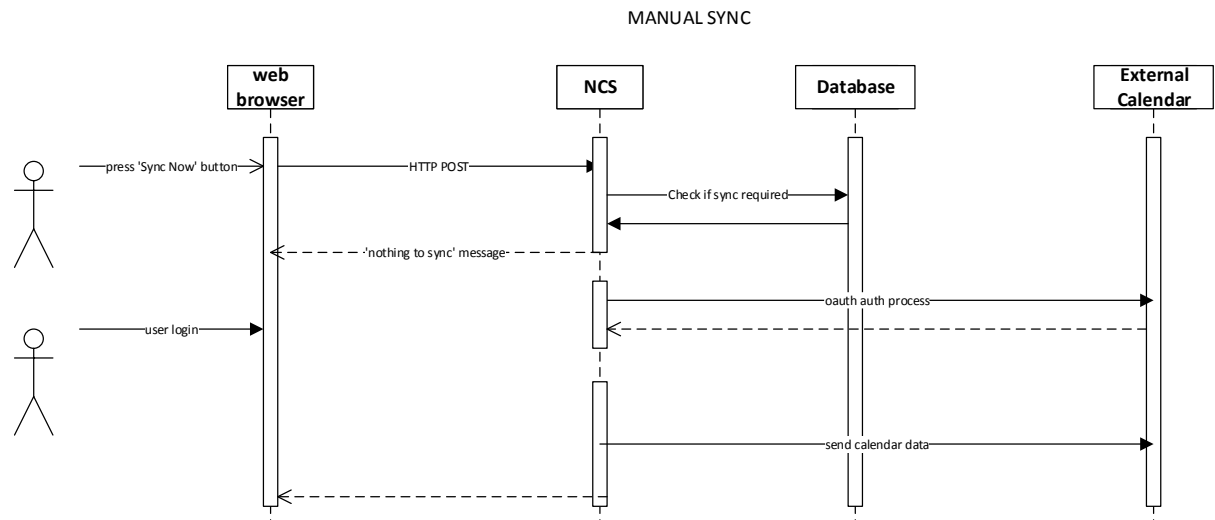
Depending the settings, appointment data from external server will be pulled from the external server and gets inserted into the Notifive database table.

5 Sequence Diagram

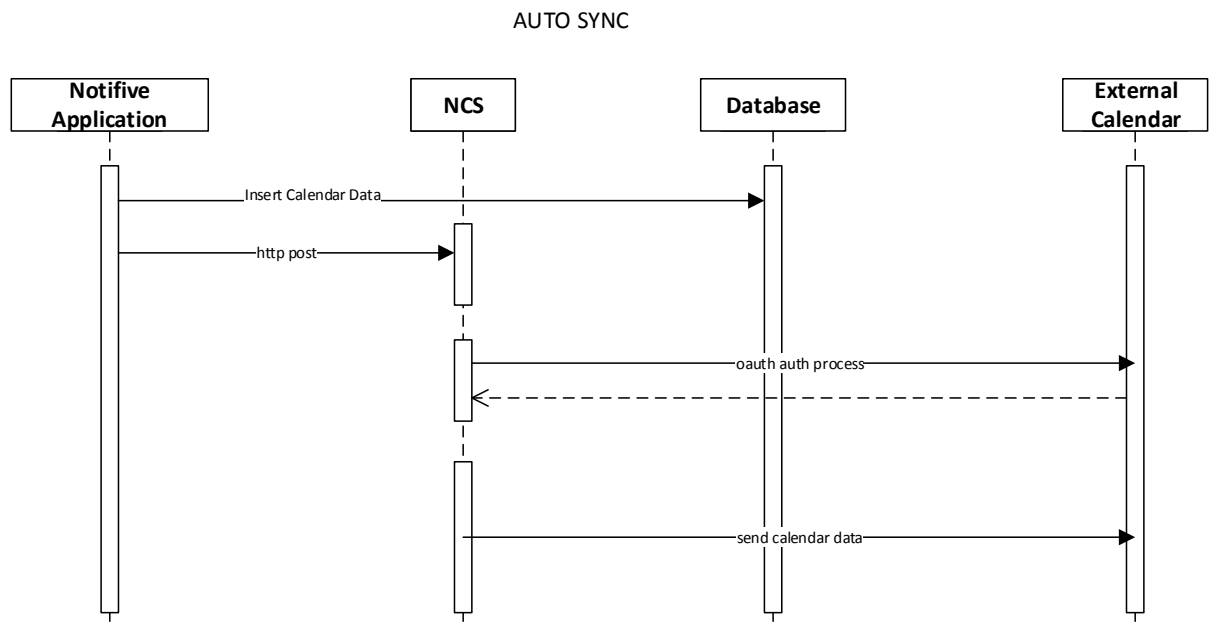
5.1 User sets / updates the settings



5.2 Manual Sync



5.3 Auto Sync



6 Table Schema

6.1 table_appointment

```
CREATE TABLE table_appointment
(
  appointment_id serial NOT NULL,
  transaction_id int,
  client_id character varying NOT NULL,
  application_id character varying NOT NULL,
  title character varying,
  name character varying,
  start_date timestamp with time zone,
  end_date timestamp with time zone,
  duration int,
  all_day boolean,
  mobile character varying,
  email character varying,
  confirmation_status character varying,
  location character varying,
  notes character varying,
  calendar_id int,
  sync_status character varying,
  set_by character varying
) WITH (
  OIDS=FALSE
PRIMARY KEY (appointment_id) ) WITH ( OIDS=FALSE );
);
ALTER TABLE table_appointment
  OWNER TO postgres;
```

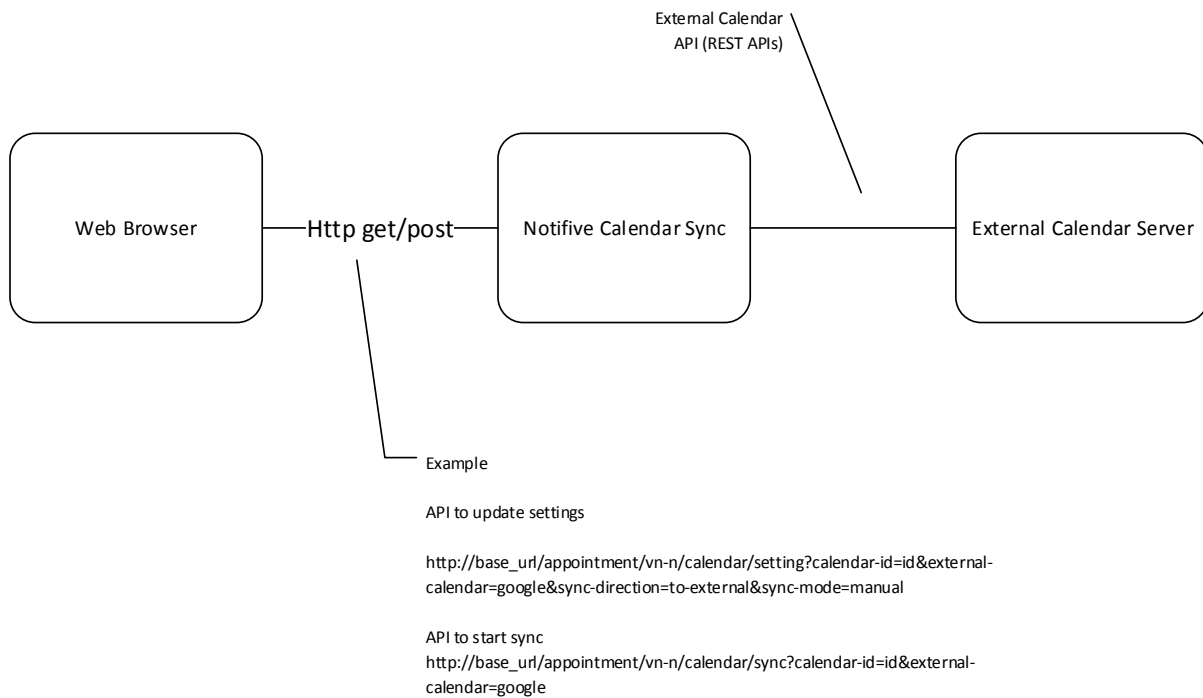
6.2 table_calendar

```
CREATE TABLE table_calendar
(
  calendar_id serial NOT NULL,
  client_id character varying NOT NULL,
  application_id character varying NOT NULL,
  name character varying NOT NULL,
  start_time character varying,
  end_time character varying,
  sync_external_calendar_type character varying,
  sync_direction character varying,
  sync_method character varying
) WITH (
  OIDS=FALSE
);
ALTER TABLE table_calendar
  OWNER TO postgres;
```

6.3 table_configuration

```
CREATE TABLE table_configuration
(
    client_id character varying NOT NULL,
    application_id character varying NOT NULL,
    timezone character varying,
    sms_two_way character varying,
    country_code character varying,
    quota_reminder_target character varying,
    quota_reminder_thresholds character varying,
    quota_reminder_levels character varying,
    quota_reminder_copy character varying
    telephone_type_tmt character varying
    international_phone_tmt character varying
) WITH (
    OIDS=FALSE
);
```

7 API



7.1 Settings

HTTP Method – GET or POST

Endpoint / URL - `http://base_url/appointment/vn-n/calendar/setting`

Query String Parameters

Parameters	Value	Description
calendar-id	Integer	Id of the notifive calendar this settings applies
external-calendar	String (Google, Apple, Office365)	Name of the external calendar system
sync-direction	String (both, notifive-to-external, external-to-notifive)	Defines how the direction of the synchronization
sync-mode	Auto, Manual	Defines the mode of the sync

Example

`http://base_url/appointment/vn-n/calendar/setting?calendar-id=id&external-calendar=google&sync-direction=to-external&sync-mode=manual`

7.2 Sync

HTTP Method – GET or POST

Endpoint / URL - http://base_url/appointment/vn-n/calendar/sync

Query String Parameters

Parameters	Value	Description	Notes
calendar-id	Integer	Id of the notifive calendar this settings applies	mandatory
external-calendar	String (Google, Apple, Office365)	Name of the external calendar system	mandatory
sync-direction	String (both, notifive-to-external, external-to-notifive)	Defines how the direction of the synchronization	Optional. Default is notifive-to-external
start-date	datetime	Calendar events from this date will be synced	Optional. If no value supplied, the current date will be used.
end-date	datetime	Calendar events to this date will be synced	Optional. If no value supplied, synchronises one month from the current date

Example

http://base_url/appointment/vn-n/calendar/sync?calendar-id=id&external-calendar=google&sync-direction=both

7.3 iCalendar

HTTP Method – GET

Endpoint / URL - http://base_url/calendar/vn-n/icalendar

Query String Parameters

Parameters	Value	Description	Notes
calendar-id	Integer	Calendar id	mandatory
appointment-id	Integer	Appointment-id	mandatory
output-format	file or string	If the output-format is 'file', api will serve the generated string as a file with extension .ics	Optional. Default is file

Example

http://base_url/calendar/vn-n/icalendar?calendar-id=1&appointment-id=24

Response

Returns calendar file in .ics format in the response body or returns the generated string in the response body

7.4 Additional Note:

You may use ***XMLHttpRequest*** or Ajax libraries which provides this functionality.

8 Testing

Tests with external servers can be done using test accounts.

Appointment database table can be populated with appropriate data for testing.

9 Milestones

9.1 Milestone 1

- Support Google Calendar only
- Support one way synchronisation only (from Notifive to Google)
- Support log4j
- Support Setting page
- No JSP code. Only static HTML +CSS + JS.
- All configurable parameters should be in web.xml or database tables (config table)
- Only manual sync required (using 'sync now' button)

9.2 Milestone 2

- Support two way synchronisation. Calendar events synchronized from google needs to be marked as 'synced'
- Calendar events created in the database should set the 'set_by' field with the external-calendar name
- Support iCalendar API as shown in section 7.3
- Improve Sync method to support start-date and end-date
- Support time zone. Time zone is provided in the table_configuration.

9.3 Milestone 3

Will be provided in v0.3 after milestone 2 completion