

# Getting Started Guide

## Contents

1. Using Mashup Factory
2. Send SMS!
3. User

## Using Mashup Factory Beta

Mashup Factory Beta is an experimental interactive development environment in which you edit, test, deploy and manage your applications. When your application is finished, you can publish it on MF servers, where it's available for you or others to run.

Here's an introduction to various key parts and functions:

- Compose (Editor)

The editor contains a repository of categorized services and a work surface to combine the services.

You add services via drag & drop from the repository to the work surface. A short service description is given in the left corner. Services have input and output ports, you connect services via linking input and output ports. There is a type checker which allows you to connect only correct data types, e.g. date with date. You configure certain services with configuration data, e.g. the web dialog. This service supports you to communicate with a user in a web browser and transfer the user data to other services.

When your service is composed (i.e. all input ports are connected with an output port), you can save the composition by giving it a name.

You can debug a service by going step by step through the service watching the parameters transferred between the services.

You updated with saved services.

- Install (manage composed services)

You can use the install and manage the composed services, i.e. active, stop, rename or remove services. You can allow users and user groups to consume the services. (See section User.)

You can watch the status of a service and define the period of activation.

- Test

You can try out your application whenever you want by clicking Test. You can test the service with your account or you have the option to test with a dedicated user name to consume the services.

You see the list of services after login for test and you can start the service via click on the link.

Once you're satisfied that it's ready to go, you can consume the service from the Portal (tbd) after authentication.

- User (manage users and groups)

You can define test users to try the composed services. Edit some attributes (e.g. email address, phone number, login name) of the test users and organise them to groups, if you plan e.g. survey or a conference call.

Caution: You are responsible for the data of test users, keep legal constraints concerning data security.

- Monitor

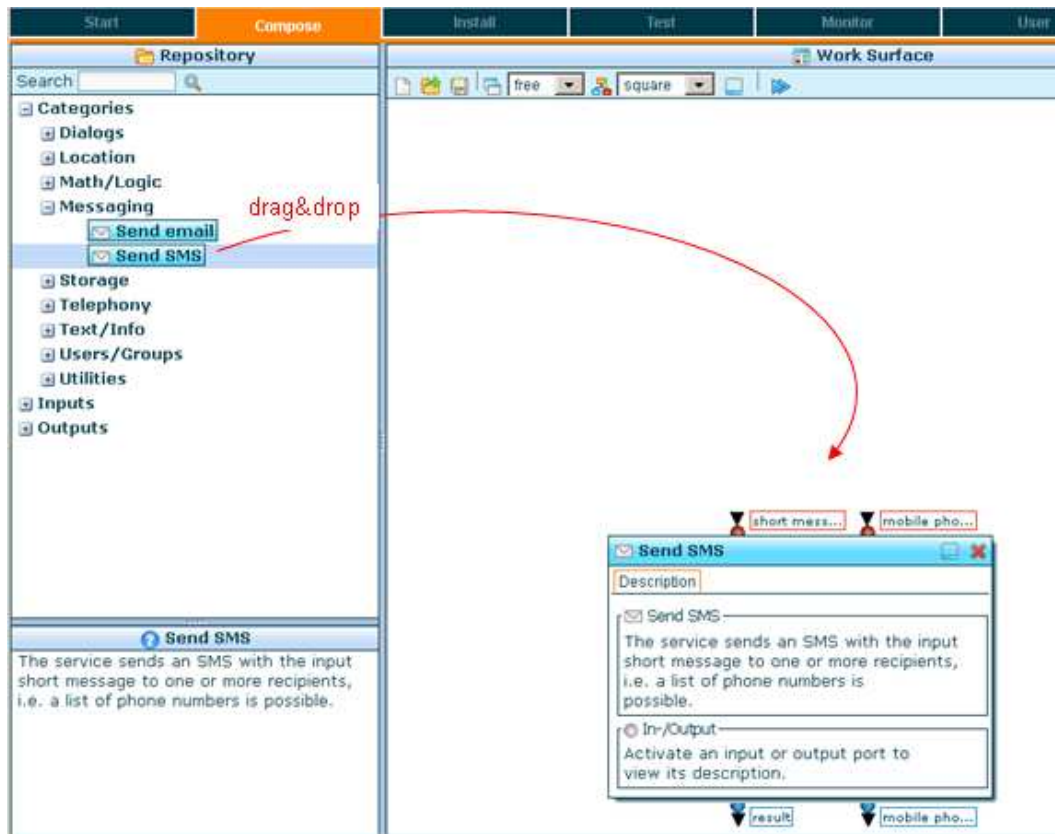
You can monitor the use some services you had executed.

## Send SMS!

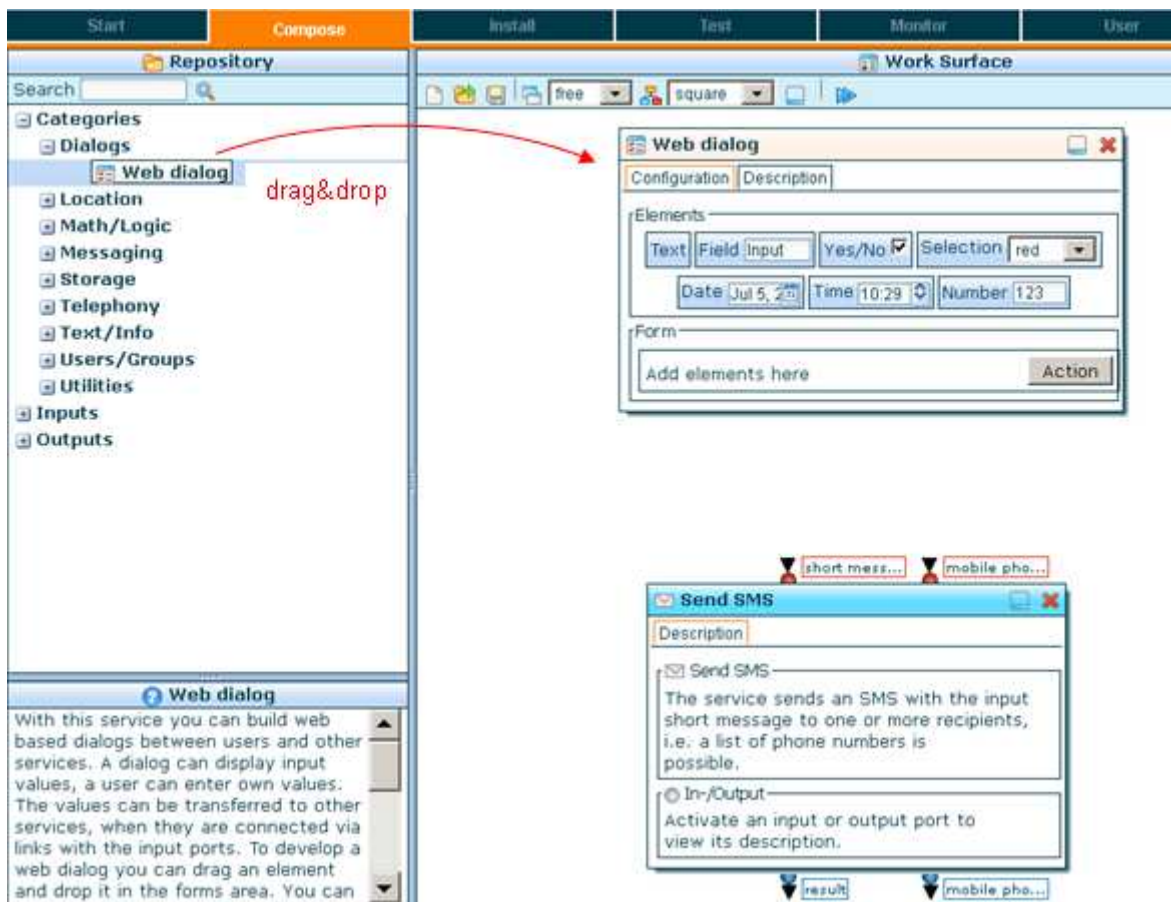
A "send SMS" service composition depicts the easy composition process with Mashup Factory and illustrates the functions and components.

Purpose of the service is a web dialog to send a SMS, which can be used by an authorized user by entering the message text and the phone number.

In the Compose section you browse the repository for “Send SMS”. Drag this box with the mouse and drop it on the work surface.

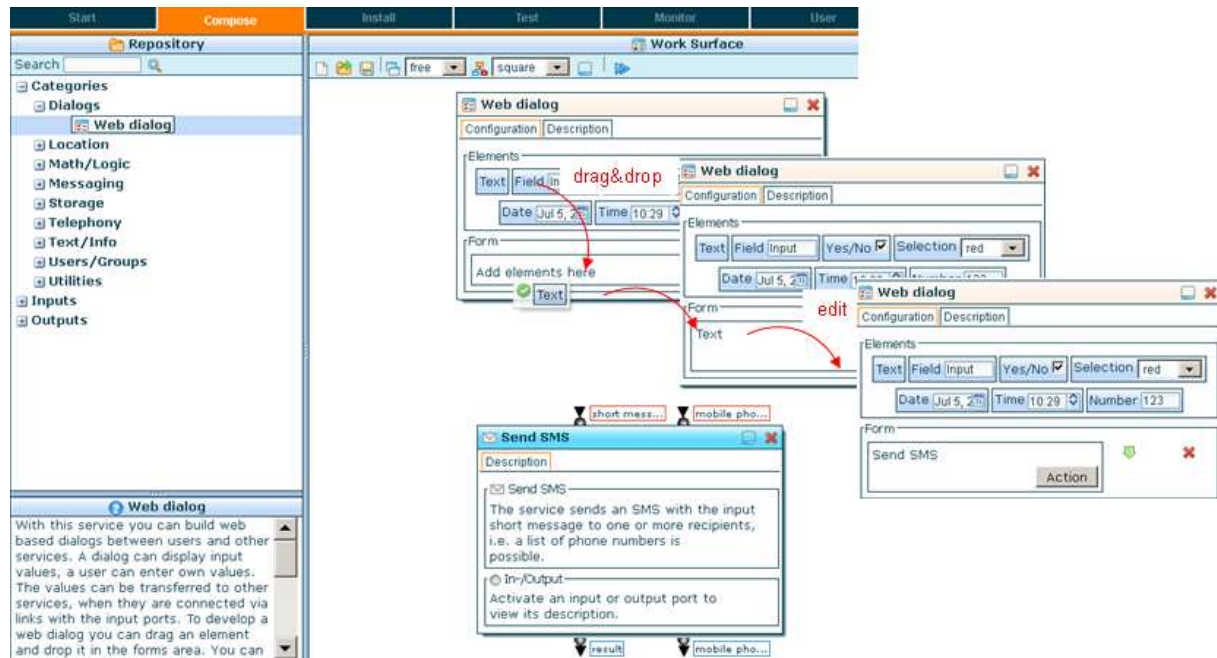


The “Send SMS” service has two input ports which have to be connected with other services. Thus, fetch the Web Dialog from the repository and drag it on the work surface.

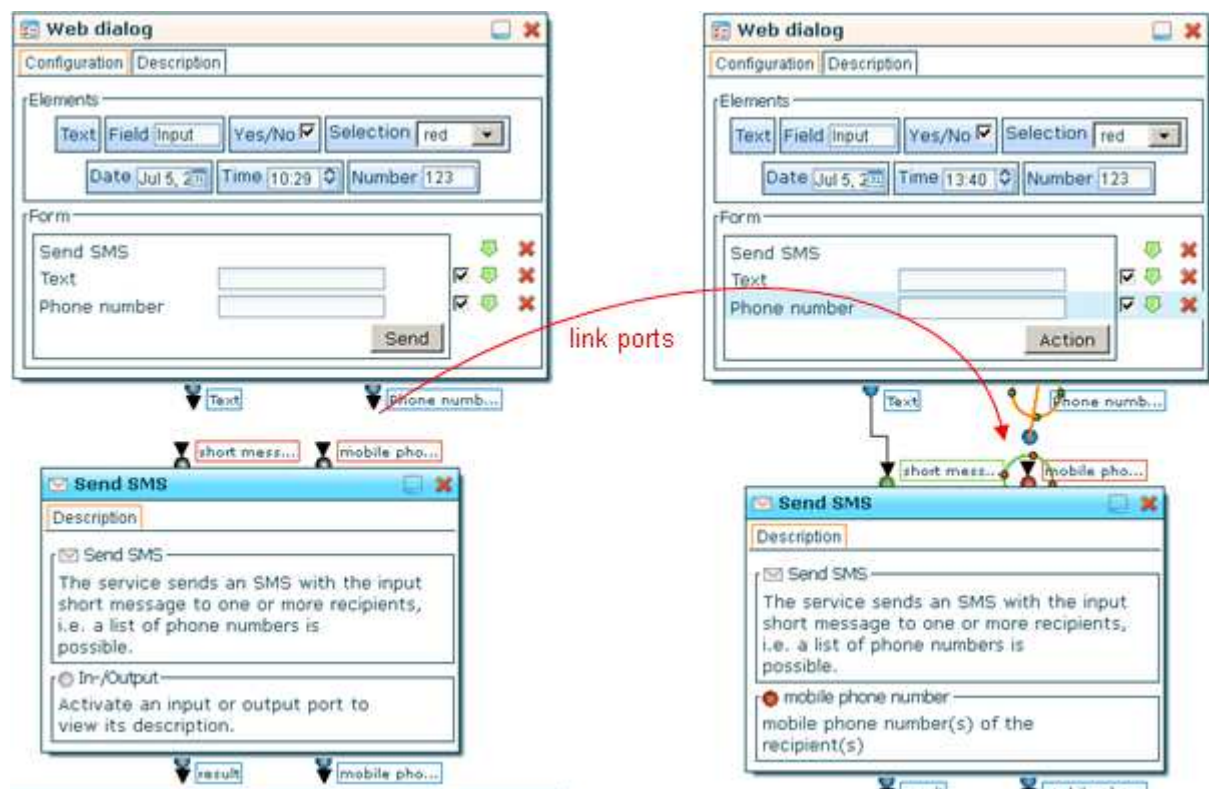


In the Web dialog you can edit the user interface which later displayed via a web browser. Relevant data fields of the user who consumes the service, are added herein.

To create the appropriate dialog you can drag an element from the Elements area of the Web dialog and drop it in the Form space. Start with dragging a text element, position it, and edit it by double clicking on the Text field. Enter “Send SMS” and press return.

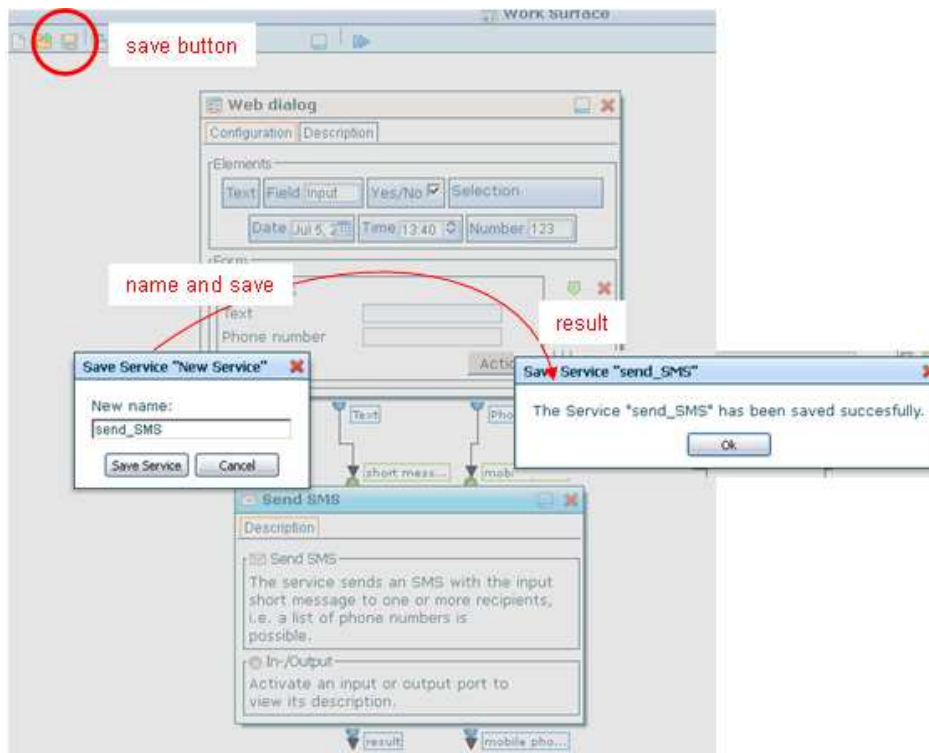


Next, enter the message text and the phone number by using the field Element. You find two new output ports below the Web dialog box. Connect the output ports with the corresponding input ports of the Send SMS.



That's it! Your first service composition is created.

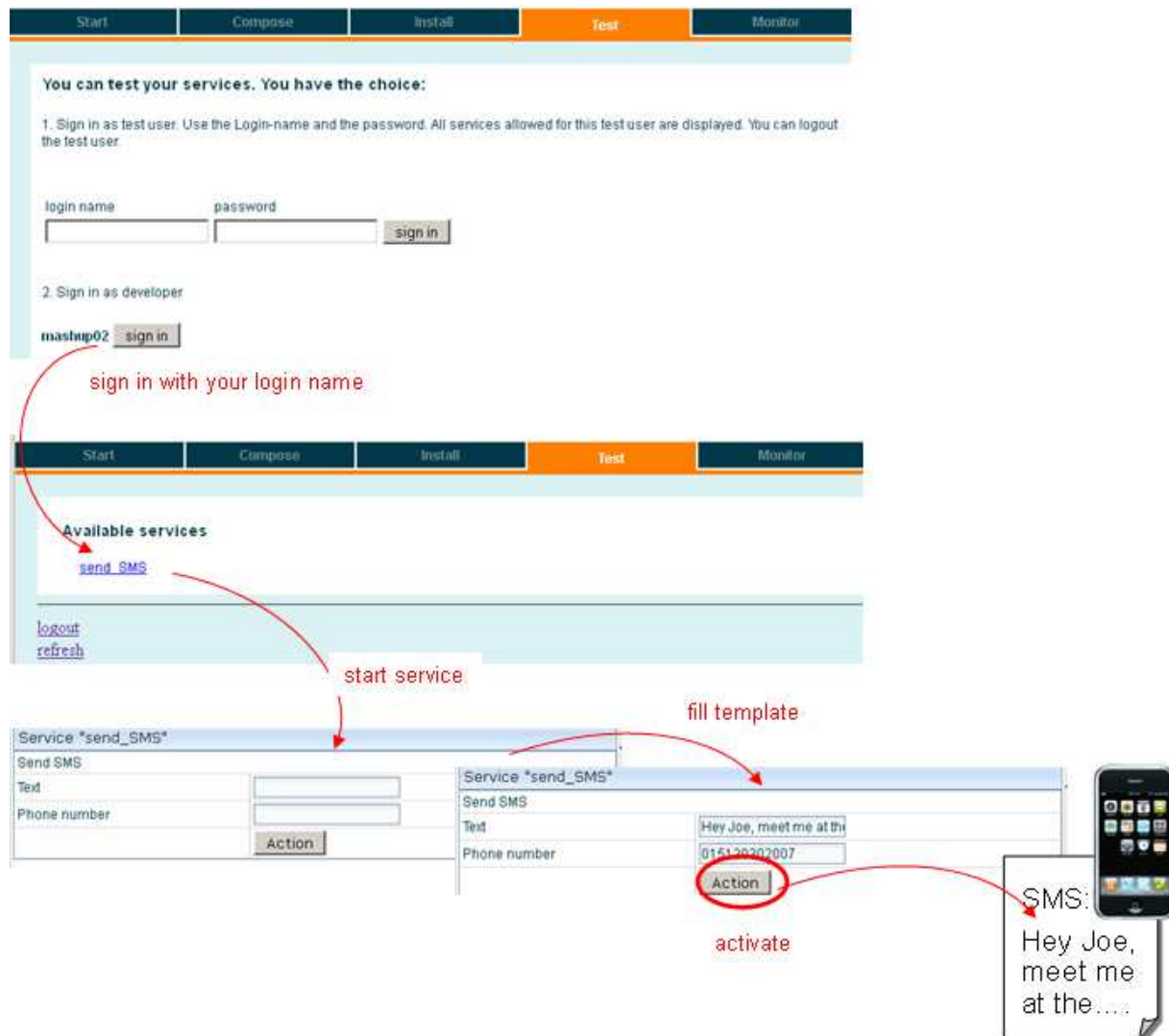
Now you save it by giving it a name. You get a result message in case of storage.



Install the new service by pushing the execution button. If the status turns to green, the service can be executed.




Test the new service and login under your user name as a developer.



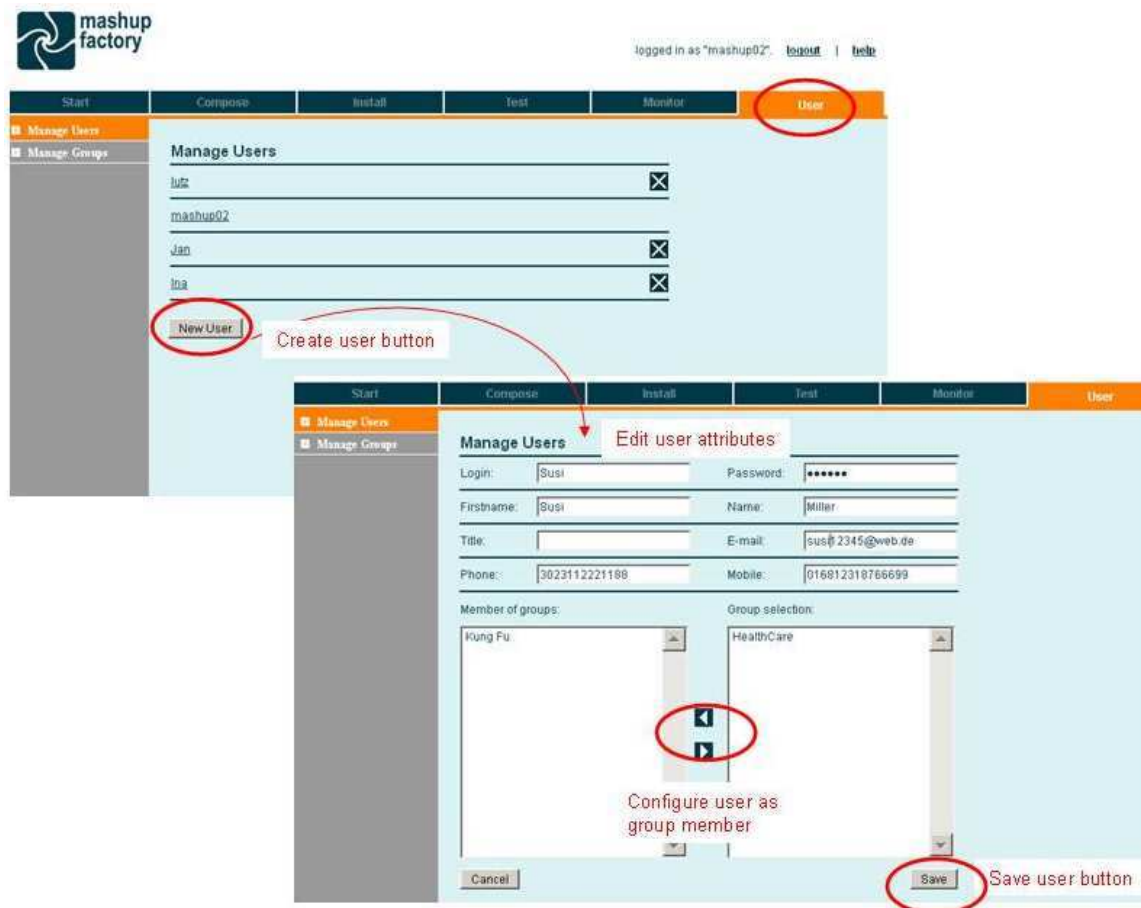
You find the new service as link and start it by pushing the link. A new tab is opened with the web dialog you have design. Fill in data, activated the action button and the SMS is sent!

## Users in Mashup Factory

You can create test users in Mashup Factory in order to do some collaborative experiments with the composed services. In the user section, go to manage users and press button 'New user'. Fill the template for the test user, don't forget a password which is necessary for authentication to consume a service. You can add a user as member to a group. The new user can be found and selected in the install section under users. You remove a user by pushing the  button.

Group management allows you to create and delete your own groups.





# Getting Started Guide – special concepts

## Contents

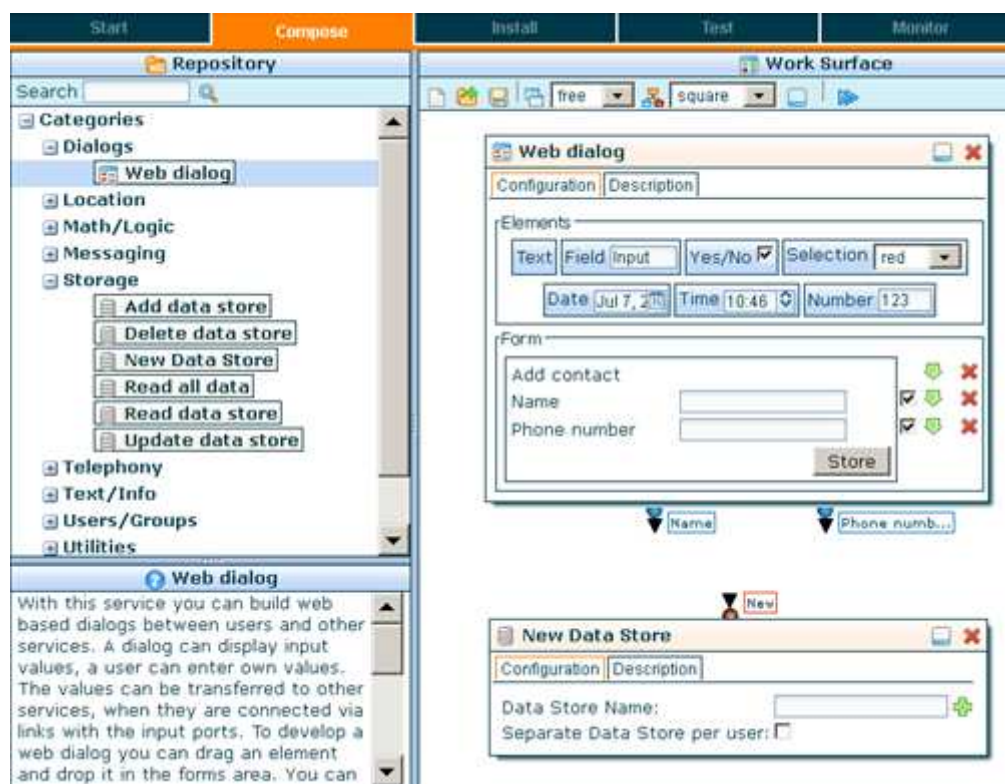
4. Using data stores
5. Setting up conferences, using user services
6. Using logical operators
7. Using google maps and geo location

## Using data stores

A data store can be used to store and retrieve data for your service composition, e.g. diary data, addresses, locations etc. Once you have created a data store you can add, modify, read or remove the data.

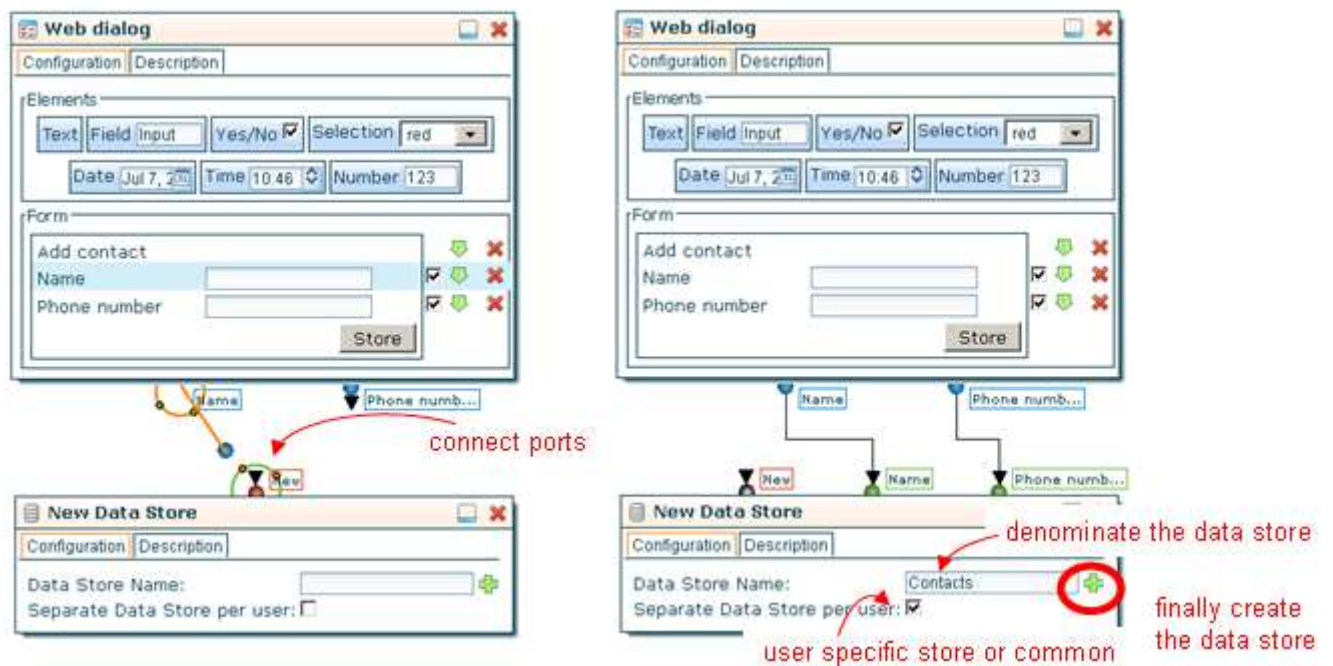
A simple addressbook with the two services (add a new contact and read my contacts) illustrates the handling.

For adding a new contact you drag the ‘New data store’ from the repository. The input port is connected with one or more output ports from other services. Here a ‘web dialog’ with a Field Element Name and Field Element Phone number are used in order to allow the user to enter the data.

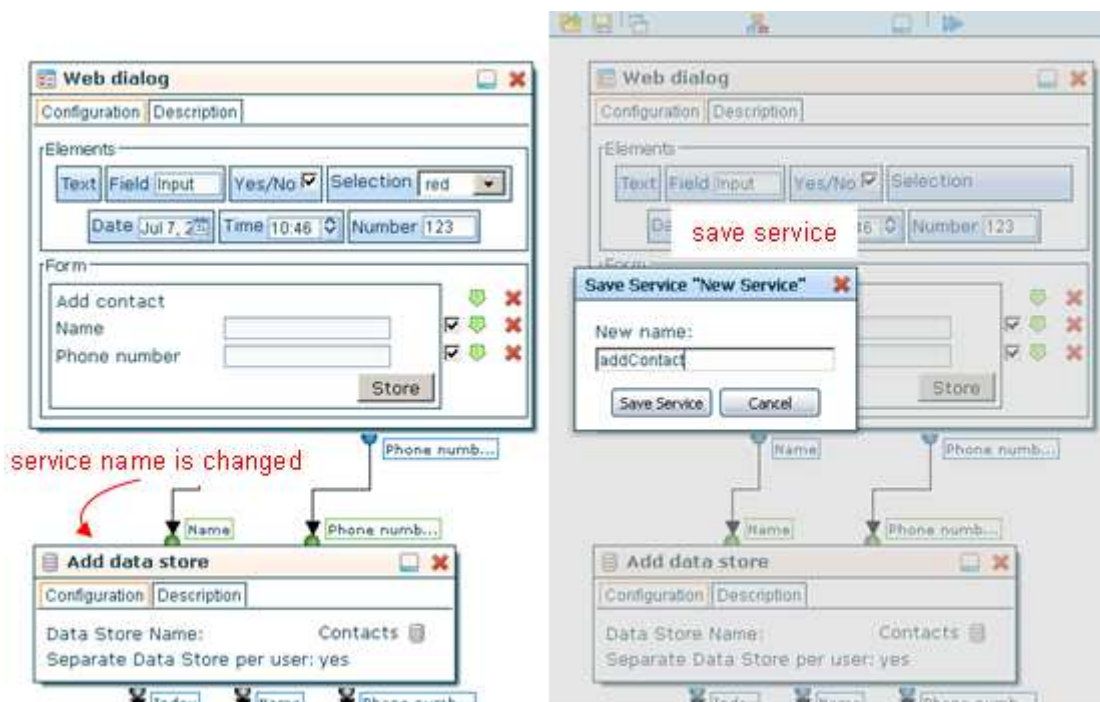


After connecting of an output port (e.g. Name of ‘Web Dialog’) to the input port New (of ‘New Data Store’), the port is added to ‘New data store’. When all input data are defined for the data store, denominate it with a name (e.g. Contacts) for the data store. You can decide, whether the data store is user-specific (e.g. diary) or common (e.g. survey) by setting a flag.



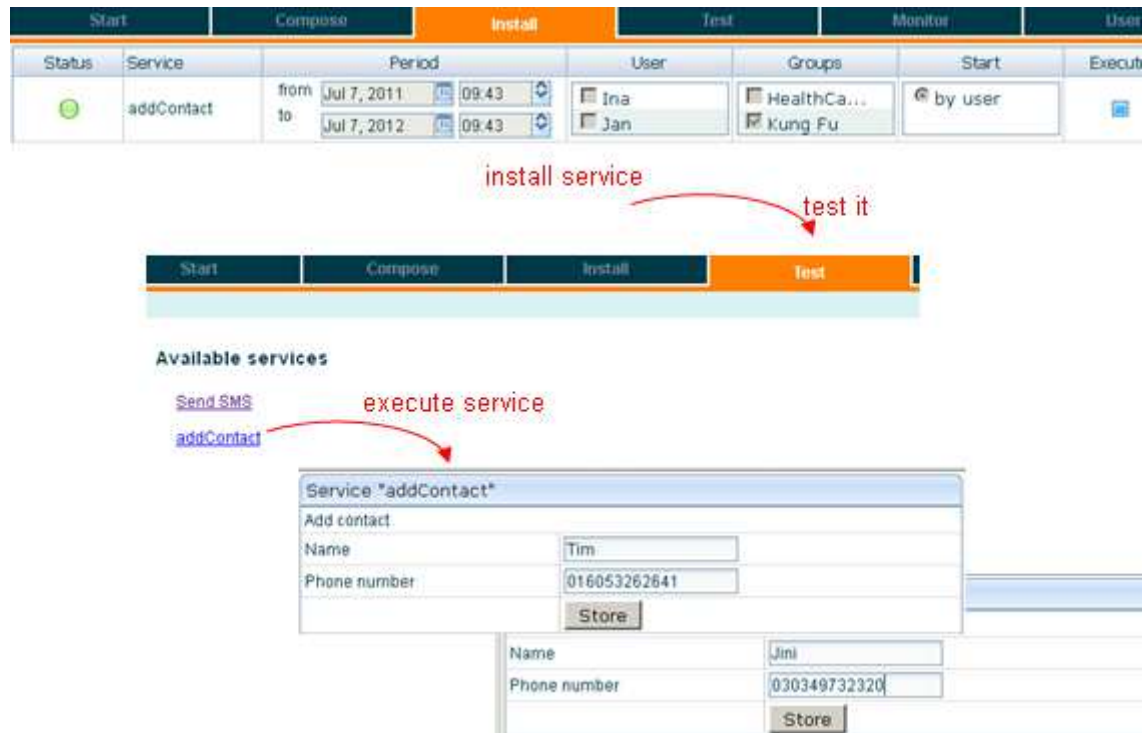


Click the + to save the store. After storage the name of the service is changed to 'Add data store'. Then save the new composition by giving it a name (e.g. addContact).



You install it and test it by clicking on the link.

Enter data in the dialog and repeat it with some contacts.

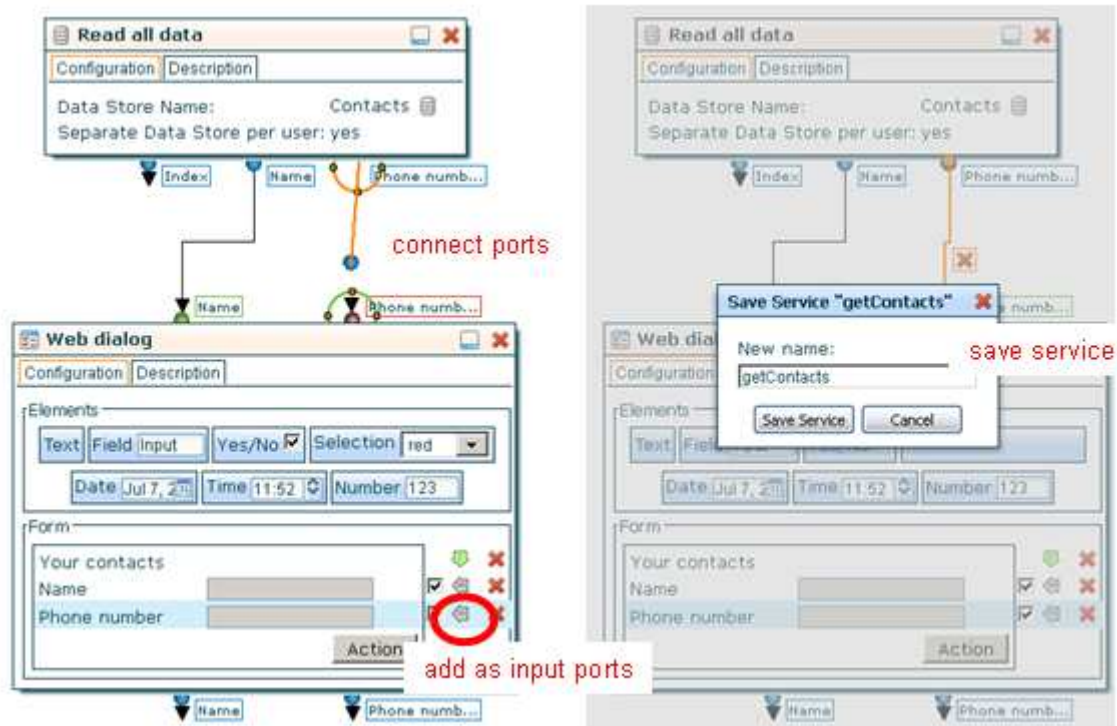


For displaying the data of your contacts, create a new service for reading the contacts.

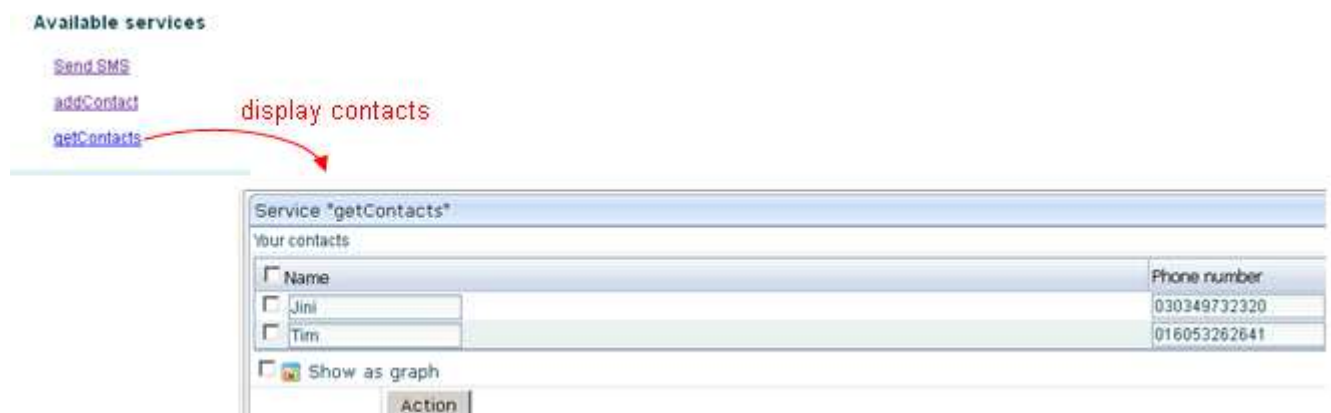
Drop the 'Read all data' service and choose Contacts from the Data Stores. The service has the output ports to read the data of Contacts (and in additional port Index, which can be used to modify the data).



To display the data you can use a 'Web dialog'. Configure the dialog by dragging and dropping the Elements. Mark the fields Name and Phone number as input ports by pressing the green arrow. Connect the corresponding ports and save the composition with the name getContacts.



Install and test getContacts. You get a list of contacts, which you have added to your contact store.



## Setting up a conference, using user services

The following section illustrates the usage of the conference service and user data.

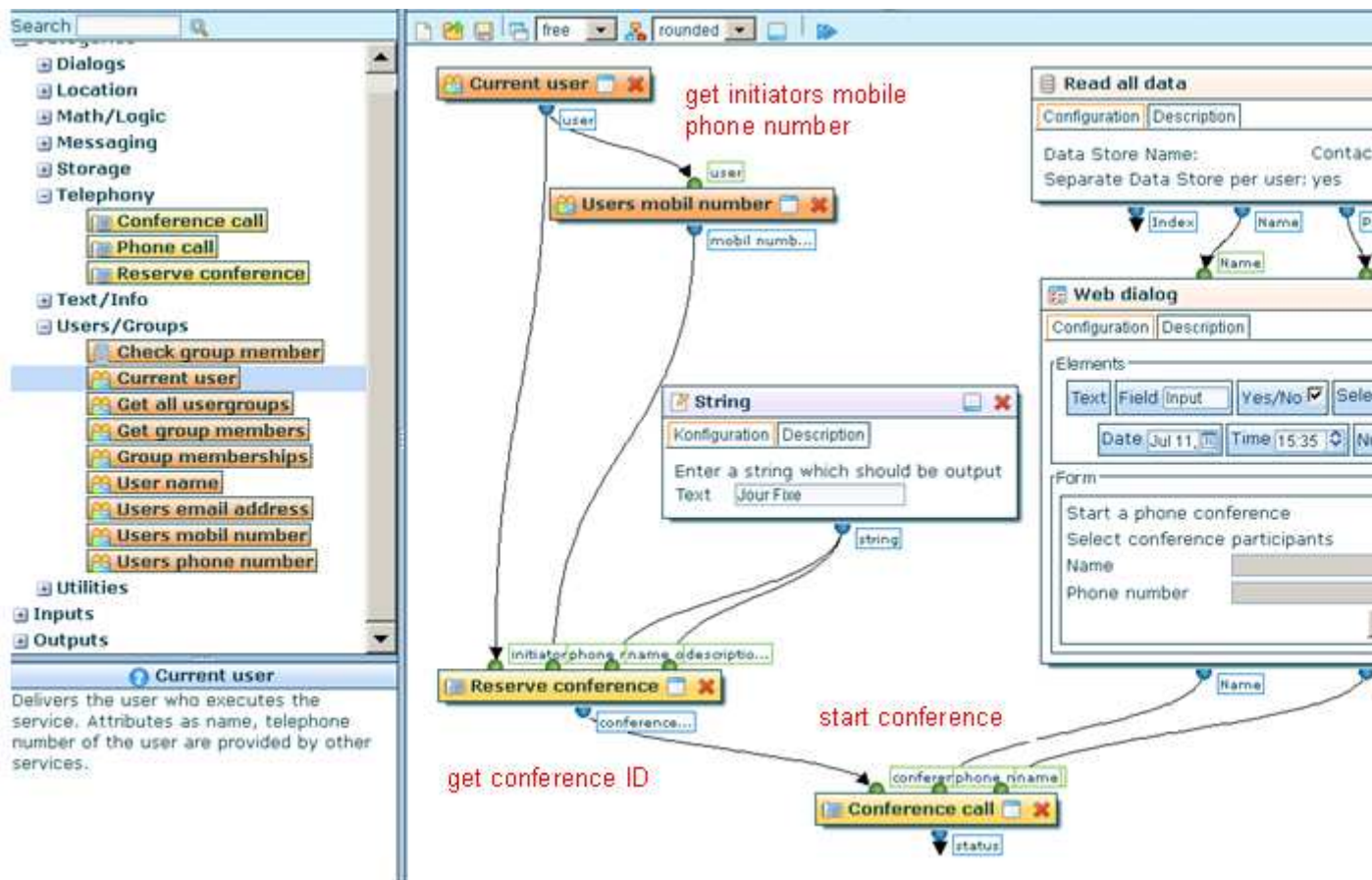
Imagine you wish to set up a call with your friends. The services 'Reserve conference' and 'Conference call' support you to start the conference. It is split in two parts to handle the conference flexible: 'Reserve conference' produces a conference id, which is used to setup a call with a set of participants. If you later on during the call decide to invite more participants, you can use this conference id to add them to the conference.

A special role in this context plays the invoker of the composed service. It can be the creator who has designed the service or any another user, who invokes the service. The user must be listed in the user management (see User of the Mashup Factory and authorised to use this service (see Install). For invoking 'Register conference' the 'Current users' phone number is taken. Look how it's realised.

Fetch 'Current user' from Users/Groups and connect it with 'Users mobil number'. Get 'String' from Utilities and define the name and description of the conference call, e.g. Jour fixe.

Drag 'Reserve Conference' from Telephony and connect the input ports with the output ports of 'Current user', 'Users mobile number' and 'String'.

In the given example the conference participants are store in a 'Data store' called Contacts (see Using data stores). The data from data store are retrieved by 'Read all data' and displayed by a 'Web dialog'.



Within the 'Web dialog' the initiator of the conference can select the participants (here Jini and Jose) of the call by setting the check box (see the following graphic). The selected names and phone numbers are used as input for the 'Conference call'. Also the conference ID from 'Reserve conference' is required as input.

After finishing the service composition you can save, install and test it.



Service "startConference"

Start a phone conference

Select conference participants

Name	Phone number
<input checked="" type="checkbox"/> Jini	030349732320
<input type="checkbox"/> Jim	016053262641
<input checked="" type="checkbox"/> Jose	015597377256

☐ Show as graph

Select participants of  
the conference

If someone else than you as service creator should be able to initiate the service, you can authorize him to use the service. You can allow single users (here Ina) or groups of users (here Kung Fu) under Install to consume the service.

Start	Compose	Install		Test	Monitor	User			
Status	Service	Period				User	Groups	Start	Execute
	addContact	from	Jul 7, 2011	09:43	<input type="checkbox"/> Ina	<input type="checkbox"/> HealthCa...	<input type="radio"/> by user		
		to	Jul 7, 2012	09:43	<input type="checkbox"/> Jan	<input checked="" type="checkbox"/> Kung Fu			
	geoposition	from	Jul 8, 2011	09:16	<input type="checkbox"/> Ina	<input type="checkbox"/> HealthCa...	<input type="radio"/> by user		
		to	Jul 8, 2012	09:16	<input type="checkbox"/> Jan	<input type="checkbox"/> Kung Fu			
	getContacts	from	Jul 7, 2011	10:46	<input type="checkbox"/> Ina	<input type="checkbox"/> HealthCa...	<input type="radio"/> by user		
		to	Jul 7, 2012	10:46	<input type="checkbox"/> Jan	<input checked="" type="checkbox"/> Kung Fu			
	Send SMS	from	Jul 8, 2011	09:25	<input type="checkbox"/> Ina	<input type="checkbox"/> HealthCa...	<input type="radio"/> by user		
		to	Jul 8, 2012	09:25	<input type="checkbox"/> Jan	<input type="checkbox"/> Kung Fu			
	startConference	from	Jul 7, 2011	14:06	<input checked="" type="checkbox"/> Ina	<input type="checkbox"/> HealthCa...	<input type="radio"/> by user		
		to	Jul 7, 2012	14:06	<input type="checkbox"/> Jan	<input checked="" type="checkbox"/> Kung Fu			

Allow users to  
use the service

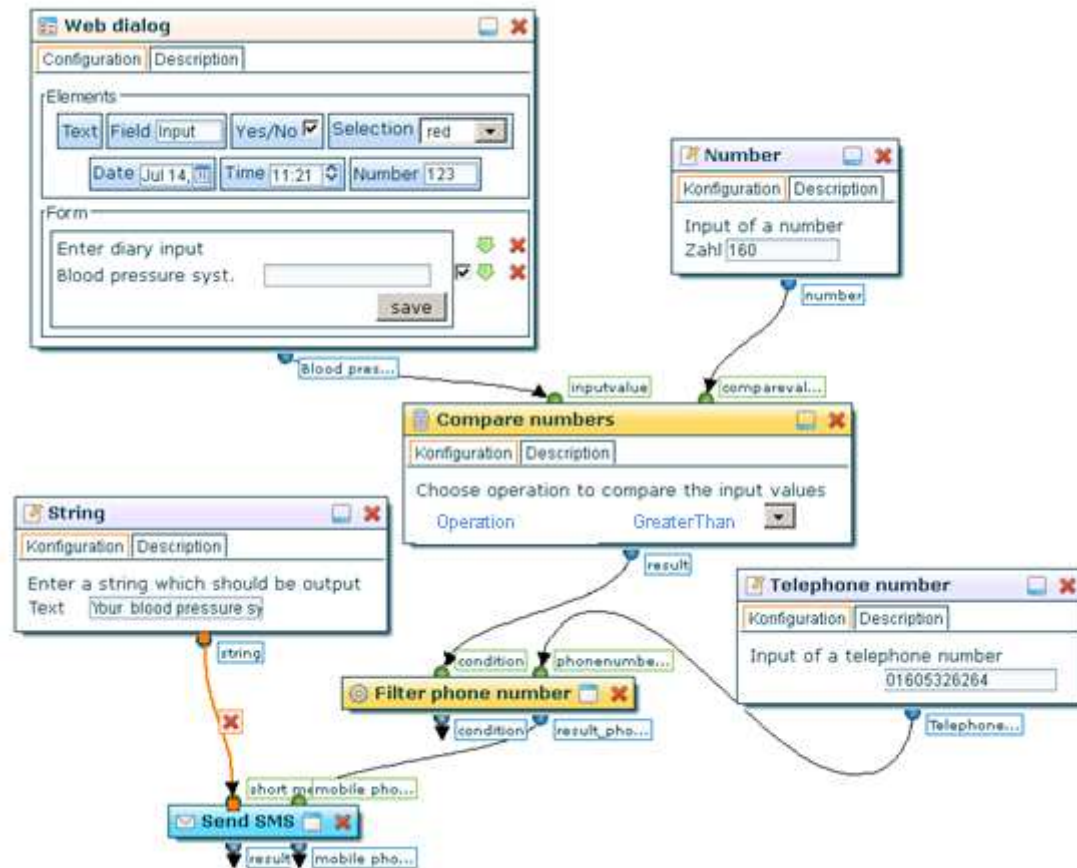
Allow all members  
in a group to use  
the service

If the Ina wants to use the service, she can do it at the web site (tbd) with her Mashup Factory account name and password.

## Using logical operators

The following section illustrates the usage of logical operators like filters and Compare operations.

You can compare a value with another value by using compare services. In the following composition (see graphic) a number value is entered by the user by a 'Web dialog'. This number is compare with a given constant number ('Number' =160). 'Compare numbers' can be configured here with GreaterThan. It results in a Boolean value (true, false), true if the input value is greater than the compare value. The result is used as input condition for 'Filter phone number'. The other input is connected with a Telephone number. If the condition is true the result of 'Filter phone number' is phone number, else the result is empty, i. e. no SMS is send.



## Geo service

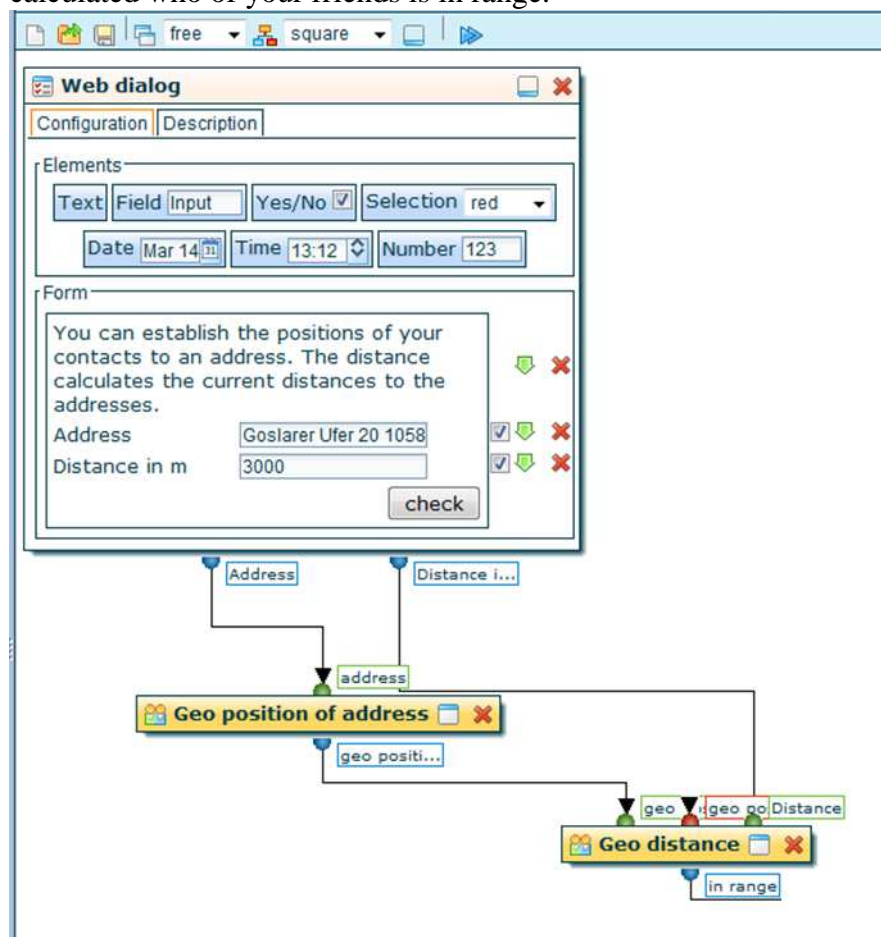
This section illustrates the handling of geo services.

Imagine you want to know who of your friends is near to your position and write a message to them.

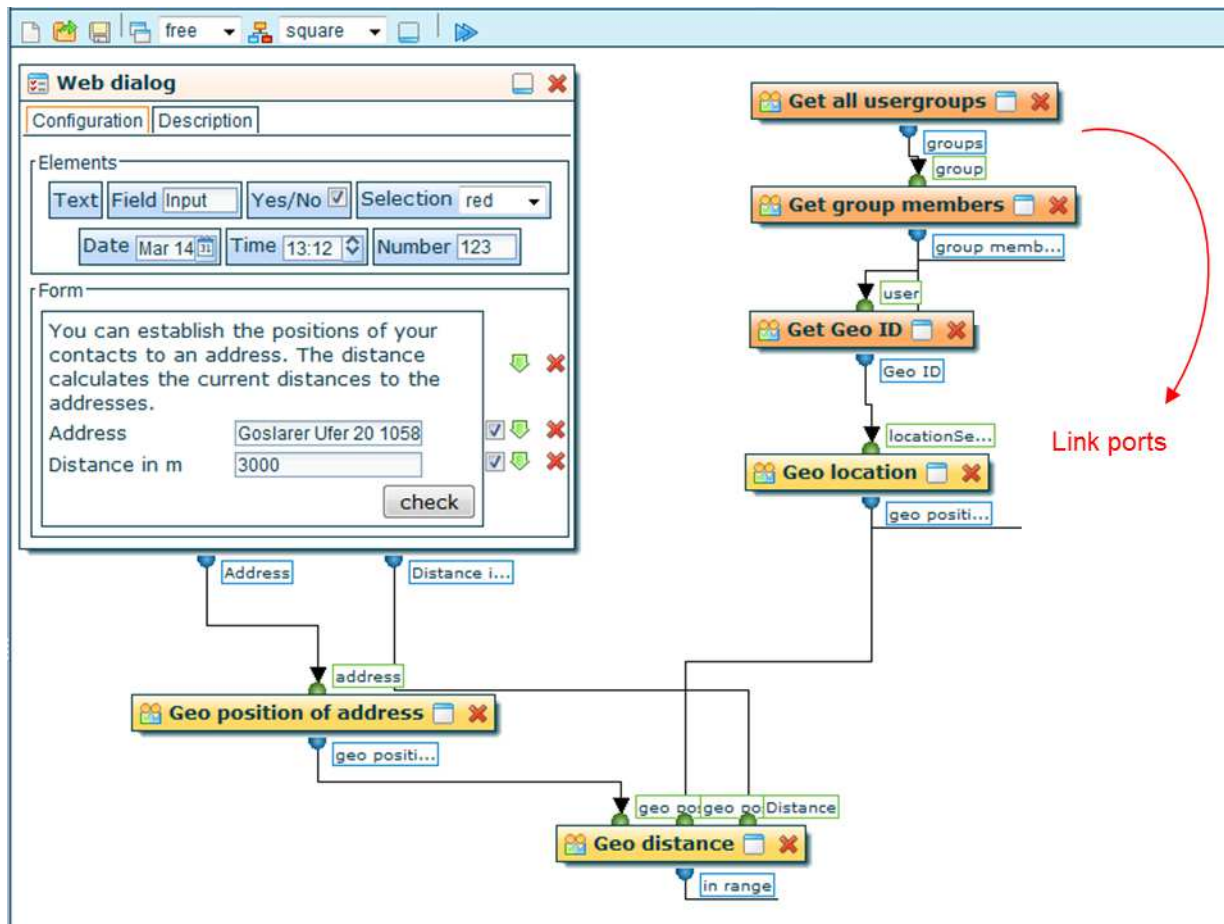
The services “Geo distance“, „Geo location“ and “Geo position of address” will support you.



Use the web dialog to specify your address and a radius (in meters). Within this radius it is calculated who of your friends is in range.



Link the output ports with the services „Geo position of address” and “Geo distance”.



Now drag the services „Get all usergroups“, „Get group members“, „Get Geo ID“ and „Geo location“ and drop them on the work surface and link the ports like in the image. The first two services allow getting your friends from the user groups. With „Get Geo ID“ you can find out where your friends are at the moment by using „Geo location“. For testing you can use this ID: 76842969699374055. The position is read through „Geo location“. This position, your position and the radius are the inputs for „Geo distance“. This service checks if the positions are within the radius and delivers a Boolean value (true, false).

The screenshot displays the 'Manage Users' interface within the Mashup Factory Beta application. The top navigation bar includes tabs for 'Start', 'Compose', 'Install', 'Test', 'Monitor', and 'User'. The left sidebar shows 'Manage Users' and 'Manage Groups'. The main content area is titled 'Manage Users' and contains the following form fields:

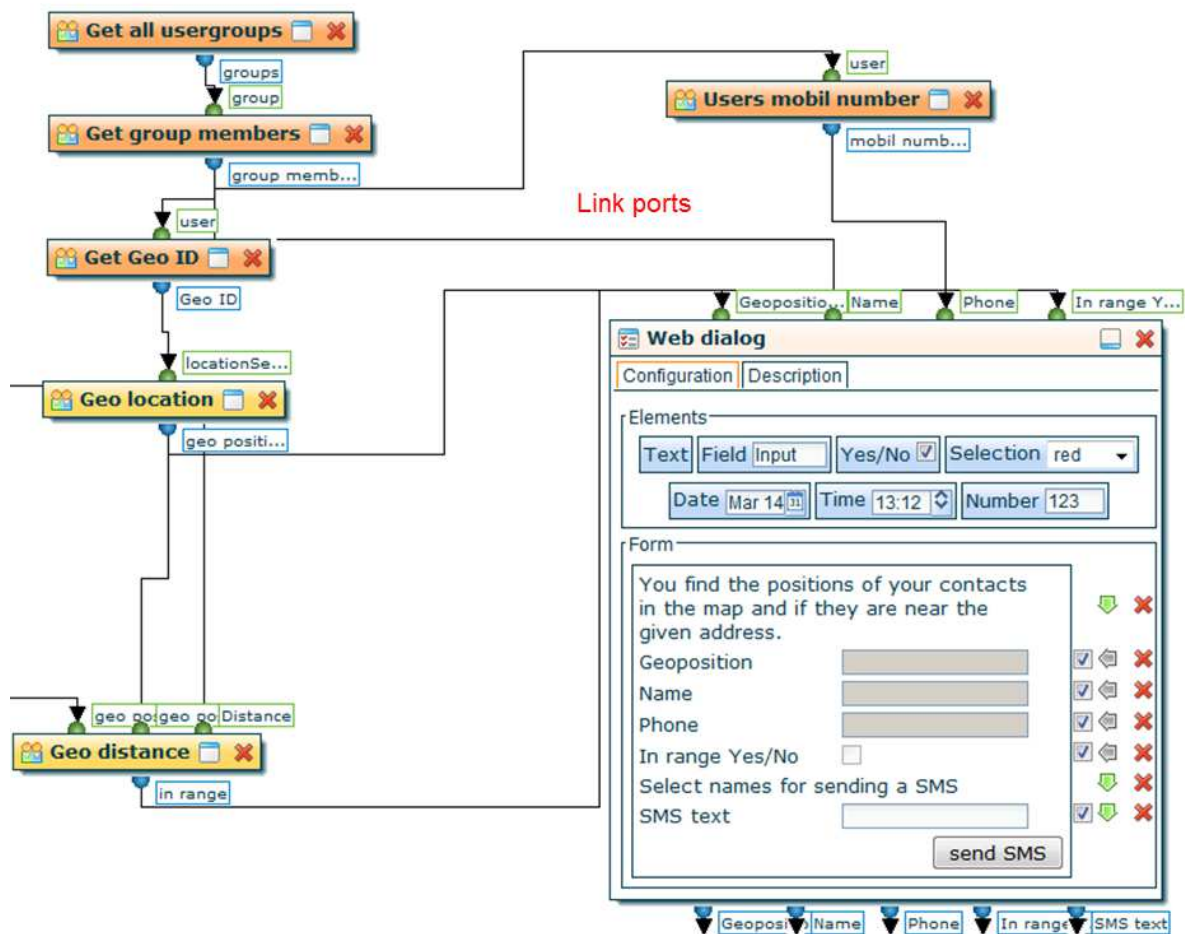
- Login: rolfender08
- Firstname: Rolf
- Name: Ender
- Google Badge ID: 1234567890 (highlighted with a red circle)
- E-mail:
- Phone:
- Mobile:

Below the input fields, there are two list boxes:

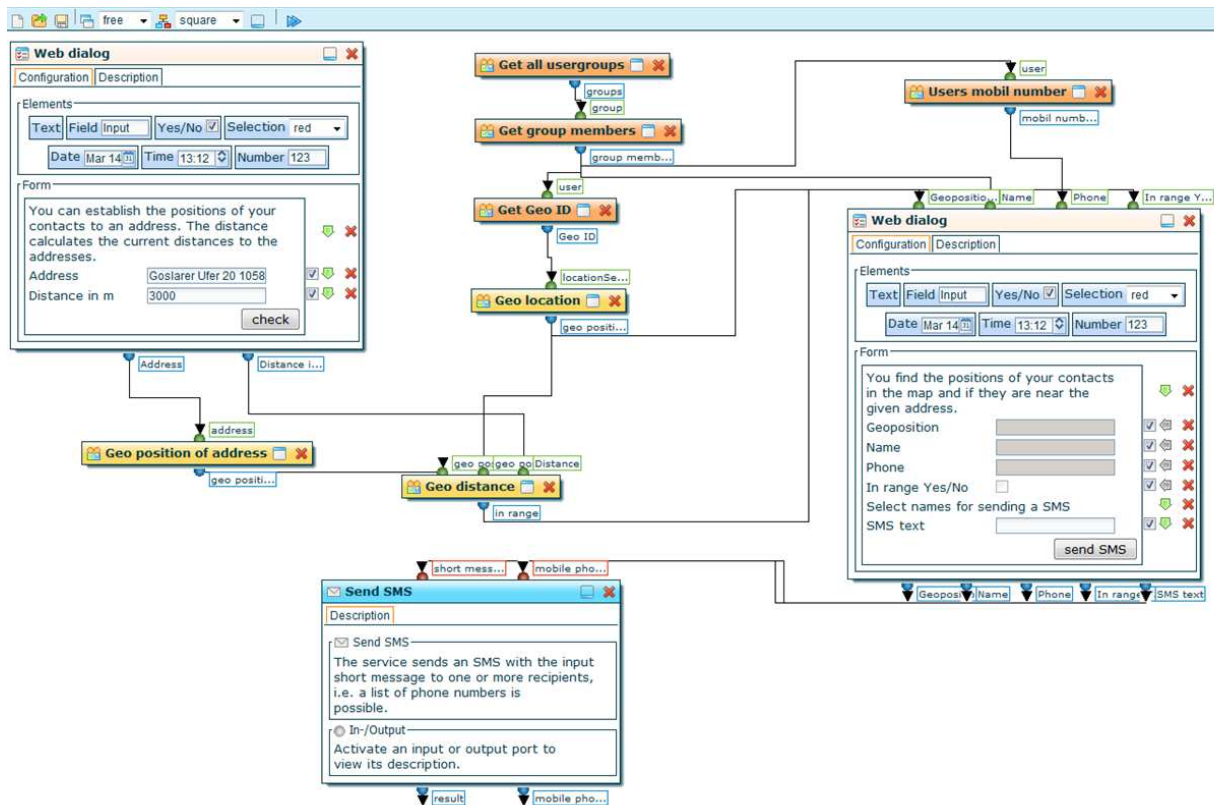
- Member of groups: KungFu
- Group selection:

Navigation arrows (left and right) are positioned between the two list boxes. At the bottom of the form, there are 'Cancel' and 'Save' buttons.

The Geo ID is essential to appropriate the current positions of your friends. The ID has to be entered into the user profiles to get the service work properly. Tip: If nothing is shown, it can be due to an invalid Geo ID or the location has not been updated in the last time. Where you can get this ID is shown in the next section.



Now add the services „Users mobil number“ and another web dialog to your composition. The web dialog shows who of your friends stands in the radius. Here the web dialog is configured to let you choose from the list of your friends to whom you send a message. You can write and send the message within the web dialog. Attention: The order of “Geoposition” and “Name” is important for displaying the positions on the map.

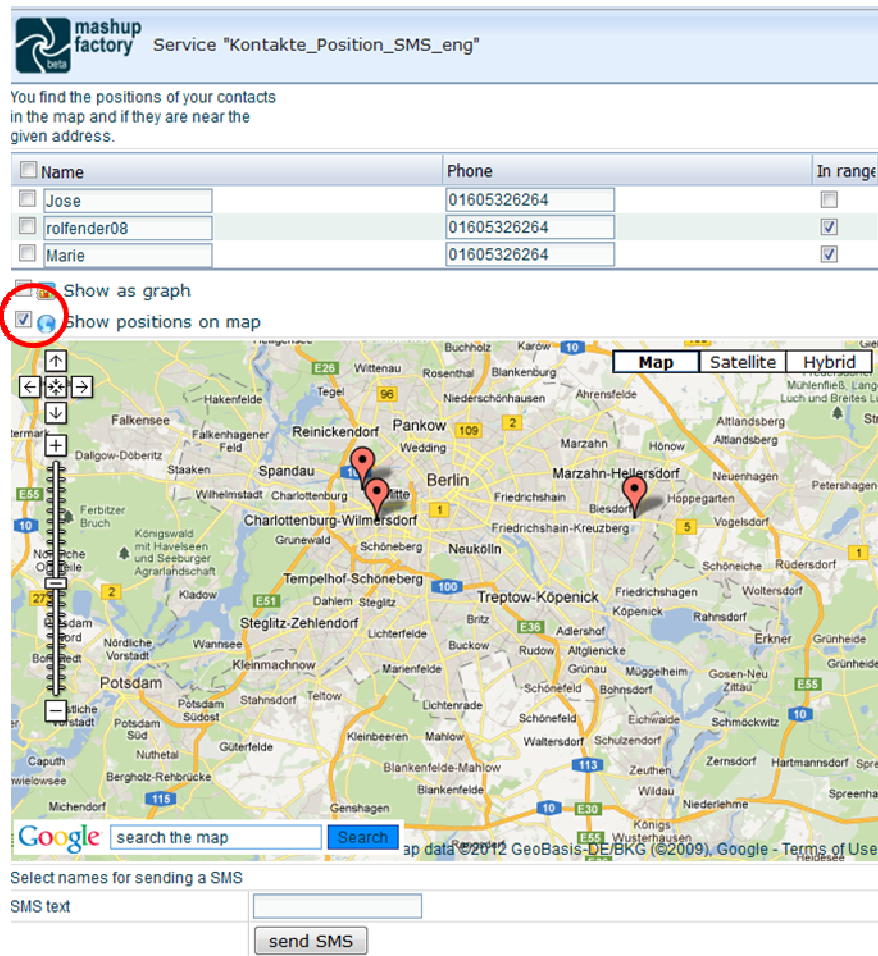


You are done. Now install and test the service.

The screenshot shows the Mashup Factory interface for the 'Service "Kontakte\_Position\_SMS\_eng"'. The interface includes a form for address input and a table of results. The table has columns for Name, Phone, and In range. The 'In range' column has checkboxes for each contact. The 'Name' column is circled in red.

Name	Phone	In range
Jose	01605326264	<input type="checkbox"/>
loffender08	01605326264	<input checked="" type="checkbox"/>
Marie	01605326264	<input checked="" type="checkbox"/>

With setting the flag you can choose to whom you want to send the message.



The screenshot shows the Mashup Factory interface for the service "Kontakte\_Position\_SMS\_eng". It includes a table of contacts with their names, phone numbers, and a checkbox for "In range". Below the table, there are two checkboxes: "Show as graph" and "Show positions on map", with the latter being checked and circled in red. The main area displays a Google Map of Berlin with red pins indicating the locations of the contacts. At the bottom, there is a search bar and a "send SMS" button.

Name	Phone	In range
Jose	01605326264	<input type="checkbox"/>
rolfender08	01605326264	<input checked="" type="checkbox"/>
Marie	01605326264	<input checked="" type="checkbox"/>

☐ Show as graph  
☒ Show positions on map

Select names for sending a SMS  
 SMS text:

If you set the flag at „Show position on map“ you can see where your friends are in the moment.

## Using Google\* maps and geo location

One of the builtin features of the dialogue basic service is its capability to display geo positions with Google maps. MashupFactory offers two services to get such geo positions:

- use 'GeoPosition' to get geo coordinates for the given address from Google maps, and
- use 'GeoLocation' to get geo coordinates for a Google user.

\*Google maps and latitude are products of google corp.

For privacy reasons the GeoLocation has been secured by Google. To get geo coordinates for a Google user, the following preconditions must be fulfilled:

- First, the Google user must have a subscription to Google latitude (the user can get it at <http://google.com/latitude>).
- Second, the Google user must have enabled Google Public Location Badge to show his location (<https://www.google.com/latitude/b/0/apps>)
- Third, the user must have entered the userid from the Google Public Location Badge code snippet into MashupFactory.

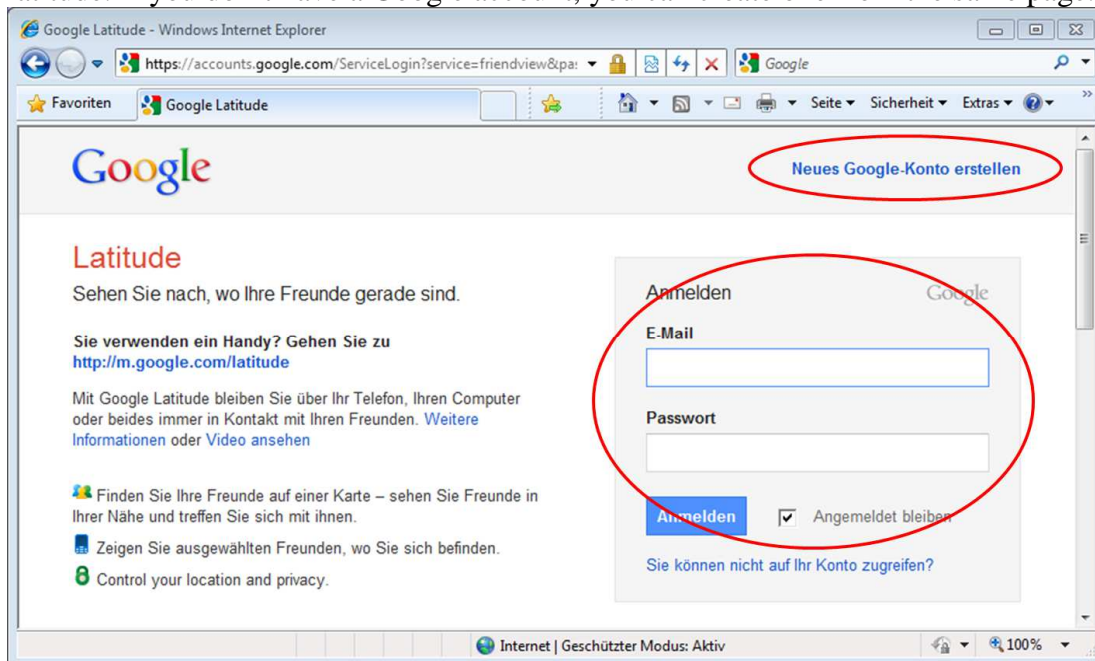


- And last but not least there should be valid location information at Google latitude (either by automatic updates from your phone or by manual setting the location on the latitude user homepage). Because without a valid location there's nothing else to do for GeoLocation than to fail ☺

These steps have to be executed by each Google user individually.

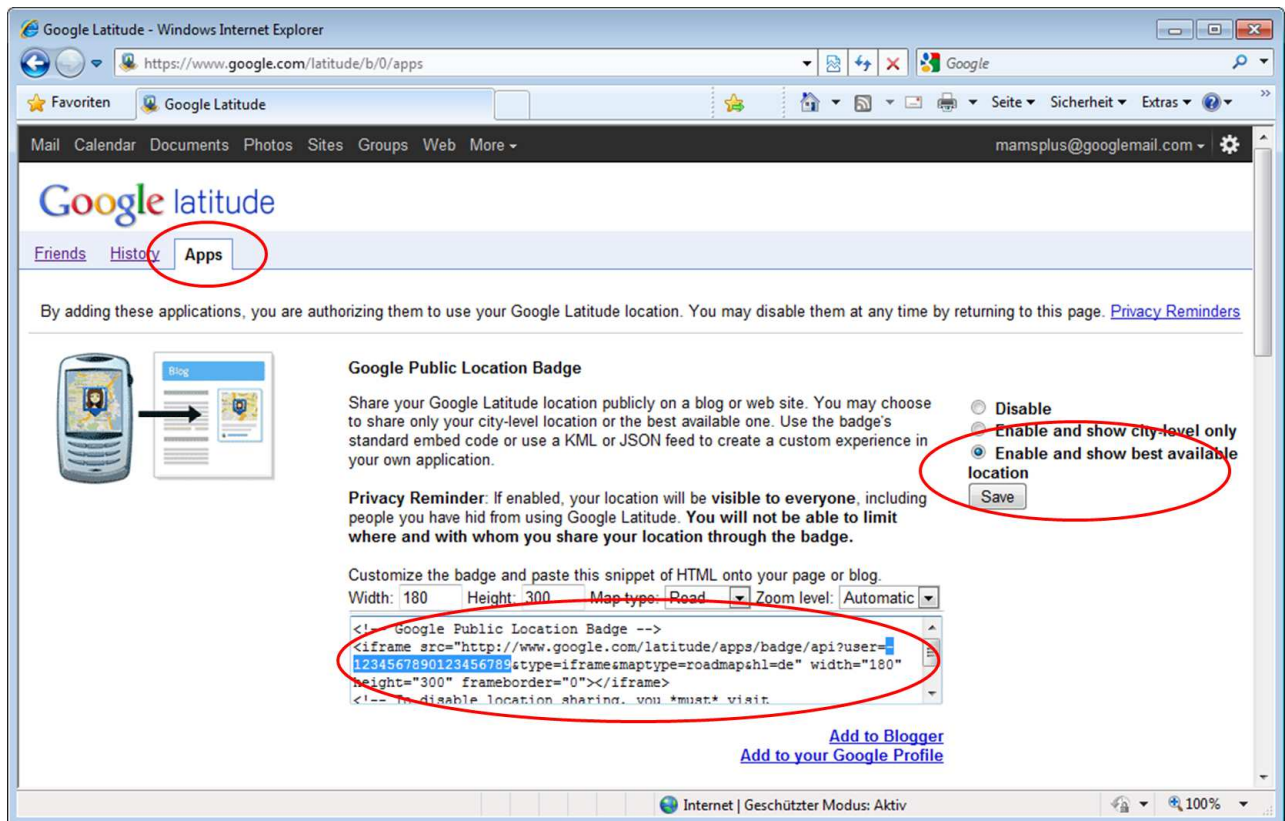
Get a subscription to Google latitude:

Go to <http://google.com/latitude> and sign in with your Google account to subscribe to Google latitude. If you don't have a Google account, you can create one from the same page.



Enable Google Public Location Badge:

After login in to latitude the Google user activates the Google Public Location Badge in the 'applications' tab.



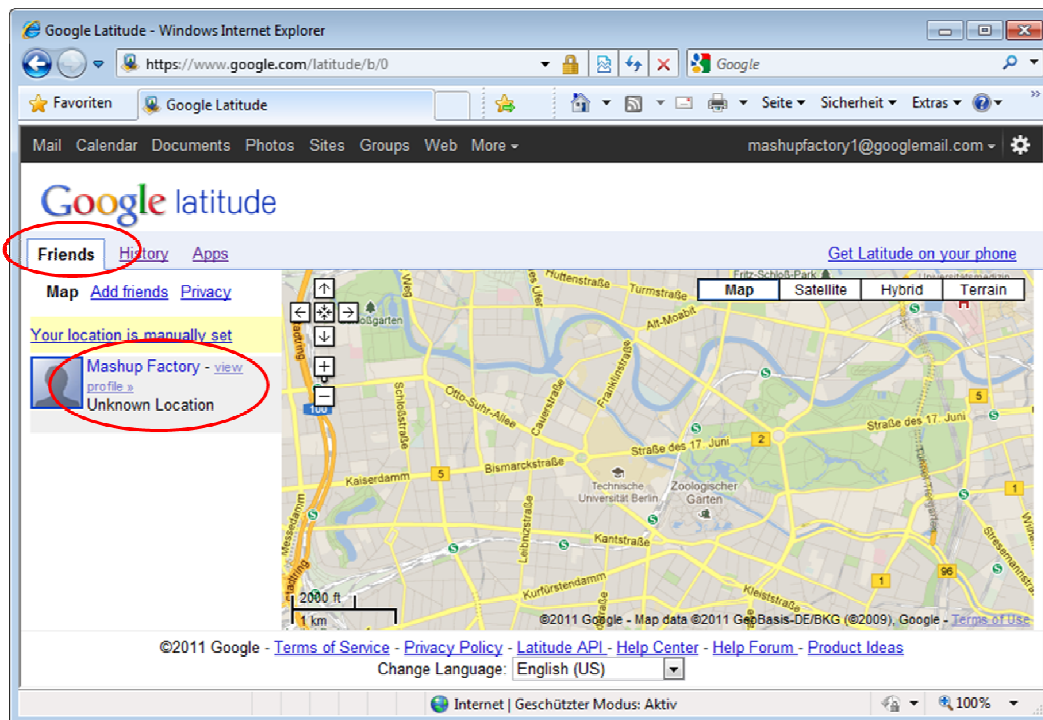
Get the userid from the Badge code snippet:

Copy the userid from the Google Public Location Badge HTML snippet; be careful: some userids start with a '-' sign, don't miss it on copying!

```
<!-- Google Public Location Badge -->  
<iframe src="http://www.google.com/latitude/apps/badge/api?user=  
1234567890123456789&type=iframe&motype=roadmap&hl=de" width="180"  
height="300" frameborder="0"></iframe>  
<!-- To disable location sharing, you *must* visit
```

Ensure there is a valid location:

Initially Google doesn't have any location information about a user, and sometimes Google will drop location information if it's completely out of date. In these cases latitude will show an 'unknown location' until the user provides a new location.



A new location can be set easily by following the link for the 'privacy' settings. Users with a GPS enabled smartphone may choose automatic updates. Manual settings always work, but the user is responsible for updating them regularly.

