

Route

Tram connection

from railway station to the Bremer Apartmenthotel:

Line 10 towards Sebaldsbrück and Brunnenstraße, there change in line 3 direction Weserwehr until Terminal

from the airport to the Bremer Apartmenthotel:

Line 6 direction Domsheide there change in line 3 direction Weserwehr until Terminal

from the university of Bremen to the Bremer Apartmenthotel:

Line 22 direction Gröpelingen to Stadter Straße, there change in the Bus Nr. 22 direction University

By car

A1 exit 55

Bremen- Hemelingen / city centre

Direction Weserstadion / airport

Follow the road. At the 2. traffic light left

Further follow the road. At the 2. traffic light right
about 300m straight ahead.

[Map](#) (1.67 Megabyte)

We have provided here a map for downloading.

Routenplaner

```
var initialized = false;  var gdMap;  var gdir;  var geocoder = null;  var addressMarker;  
function initialize() { if (!initialized && GBrowserIsCompatible()) { gdMap = new
```

```

GMap2($('gd_map_canvas')); gdMap.addControl(new GLargeMapControl()); gdMap.addControl(new
GMapTypeControl()); gdir = new GDDirections(gdMap, $('gd_directionList')); GEvent.addListener(gdir,
"load", onGDDirectionsLoad); GEvent.addListener(gdir, "error", handleErrors); var map_point = new
GLatLng(53.0625899,8.864862); var map_icon = new GIcon(G_DEFAULT_ICON); map_marker =
new GMarker(map_point, map_icon); gdMap.addOverlay(map_marker); map_icon.infoWindowAnchor
= new GPoint(0,0); GEvent.addListener(map_marker, "click", function() {
map_marker.openInfoWindowHtml("Bremer Apartmenthotel Garni **S
Hastedter Osterdeich 206
28207 Bremen"); }); map_marker.openInfoWindowHtml("Bremer Apartmenthotel Garni **S
Hastedter Osterdeich 206
28207 Bremen"); initialized = true; } } function setDirections(waypoints) { initialize(); var
filteredWaypoints = new Array(); for(var i = 0; i < waypoints.length; i++) { filteredWaypoints.push(waypoints[i]); } }
gdir.loadFromWaypoints(filteredWaypoints, { "locale": "en" }); } function handleErrors(){ if
(gdir.getStatus().code == G_GEO_BAD_REQUEST) showError("A directions request could not be
successfully parsed.\n Error code: " + gdir.getStatus().code); else if (gdir.getStatus().code ==
G_GEO_SERVER_ERROR) showError("The directions request could not be processed. Please try
again.\n Error code: " + gdir.getStatus().code); else if (gdir.getStatus().code ==
G_GEO_MISSING_QUERY) showError("The directions request is empty!\n Error code: " +
gdir.getStatus().code); else if (gdir.getStatus().code == G_GEO_UNKNOWN_ADDRESS)
showError("No corresponding geographic location could be found for the specified address. This may be
due to the fact that the address is relatively new, or it may be incorrect."); else if (gdir.getStatus().code
== G_GEO_UNAVAILABLE_ADDRESS) showError("The geocode for the given address or the
route for the given directions query cannot be returned due to legal or contractual reasons."); else if
(gdir.getStatus().code == G_GEO_UNKNOWN_DIRECTIONS) showError("The directions between
the points mentioned in the query could not be computed. This is usually because there is no route
available between the two points, or because we do not have data for routing in that region."); else if
(gdir.getStatus().code == G_GEO_BAD_KEY) showError(" The given key is either invalid or does
not match the domain for which it was given. \n Error code: " + gdir.getStatus().code); else
showError("An unknown error occurred.\n Error code: " + gdir.getStatus().code); } function
showError(error){ setErrorText(error); $('gd_errorBox').setStyle('display', 'block');
$('gd_map_canvas').setStyle('display', 'none'); } function removeError(){
$('gd_map_canvas').setStyle('display', 'block'); $('gd_errorBox').setStyle('display', 'none');
setErrorText(""); } function setErrorText(text){ $('gd_errorBox').set('text',text); } function
onGDDirectionsLoad(){ removeError(); }

```