Service Docking

1 Building services

This paper elaborates how to realize the data interaction of mobile terminal and SuperMap iServer server-side from the building SuperMap iServer service, data uploading server, downloading the data from server, querying server-side data several aspects.

1.1GettingSuperMap iServerproduct package

By Supermap offical website's technical resource center-cloudGISonline downloading to get SuperMap iServer product package.

1.2 InstallingSuperMap iServer

The detailed installation method and license configuration please see SuperMap iServer online help 's "iServer installation guide" section.

1.3 SuperMap iServer configuration and management

1.3.1 Server starting / stopping

SuperMap iServer as a Web application can be deployed to multiple Web servers, SuperMap iServer defaults to be deployed in built-in tomcat, starting tomcat can start iserver,simultaneously starting the service provided by SuperMap iServer.

In the %SuperMap iServer_Home%/bin directory, providing the batch files to start / stop SuperMap iServer server.

- **startup.bat**: In Windows system, starting SuperMap iServer server
- **startup.sh**: In Linux system, starting SuperMap iServer server
- **shutdown.bat**: In Windows system, stopping SuperMap iServer server
- **shutdown.sh**: In Linux system, stopping SuperMap iServer server

Advanced Configuration: In addition to using the above ways to start / stop iServer server, users can also use Windows service to start / stop iServer server, specifying configuration file directory and starting SuperMap iServer, the detailed steps please seeSuperMap iServer online help 's “iServer configuration and management"'s "server starting / stopping" section

1.3.2 Managing services

After SuperMap iServerservice is started, it will automatically publish default GIS service ,

Initializing configuration

When starting the service for the first time, you can enter the initial configuration wizard interface (or input address in browser:http://<server>:<port>/iserver/ to enter the initial configuration wizard interface), input the user name and password to create the administrator account, then it automatically check system environment and license information, after checking the initial configuration is completed.

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

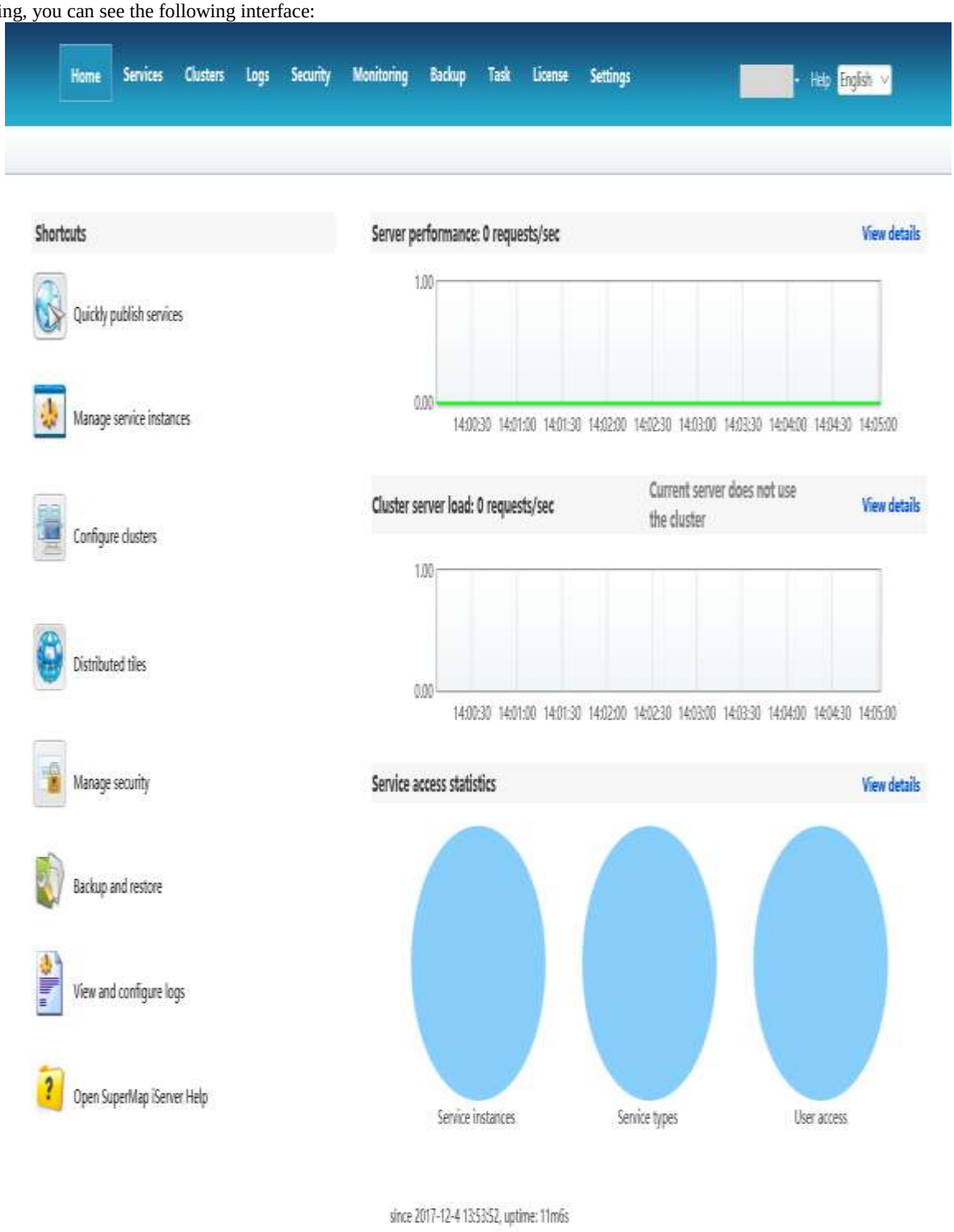
Managing service:

The administrator uses the service manager to manage the service.

Two ways of starting the service manager:

- Inputting the address as the following format in the browser address bar:http://<server>:<port>/iserver/manager/,such as: http://localhost:8090/iserver/manager/;
- Clicking "start" , "program", "SuperMap", "SuperMap iServer 8C" , "iServer service management".

Administrator inputs the user name and password creating at the first time, then uses SuperMap iServer service manager. After logging, you can see the following interface:



The service manager home page provides the entry to access and configuring service. Users can quickly publish one or a set of

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

services, manage and configure service instance, service interface, service components (sets) and service provider (sets), view and configure the server logs, can also configure cluster, and quickly view help information and so on.

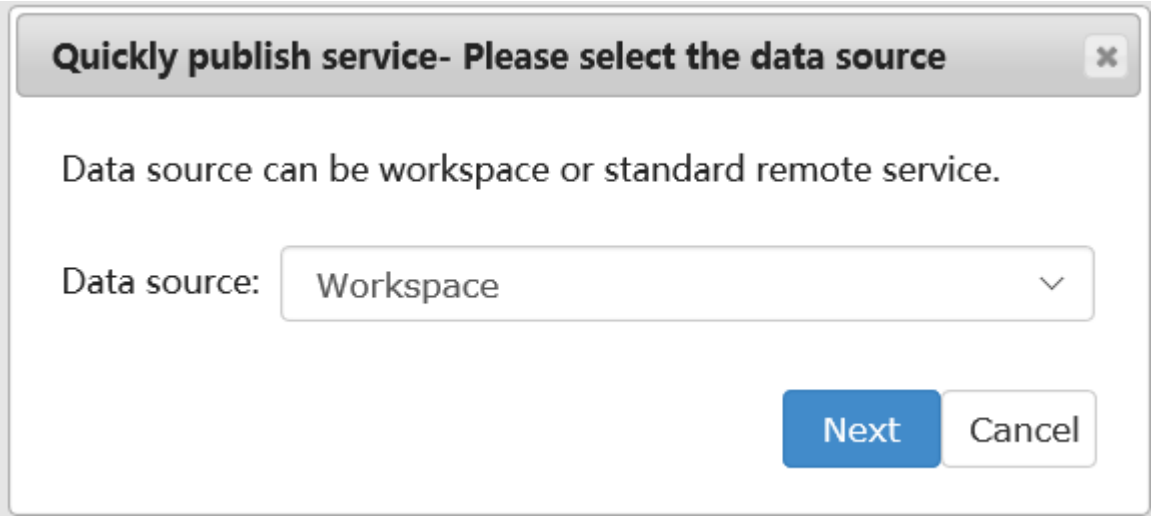
1.4 Publishing SuperMap iServer service

SuperMap iServer provides convenient quickly publishing services mechanism, supporting rapidly publishing SuperMap working space, online services and tiles package, OGC or other standard service and tiles package, third-party online map services and other GIS platform service and tiles package.

This paper takes Changchun data Changchun.smwu as an example, elaborating how to publish iServer Rest data service and map service.

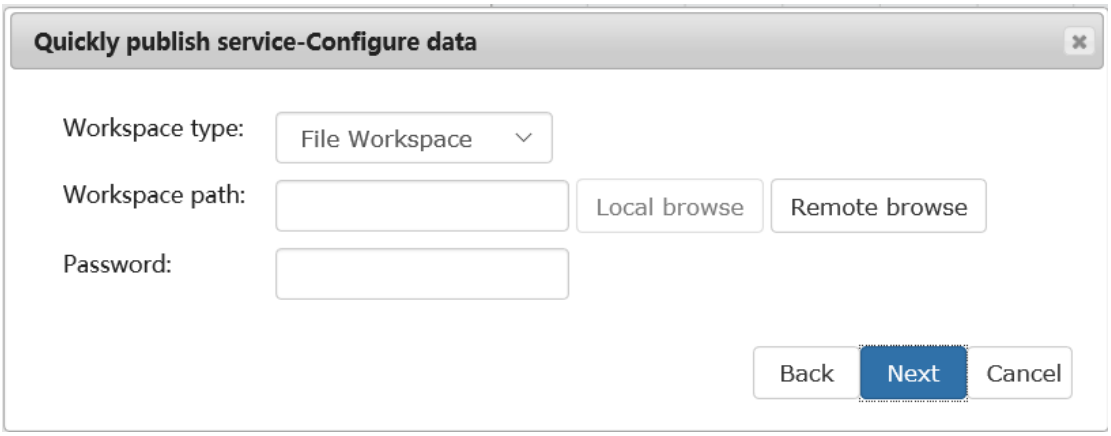
1.4.1 Selecting published resource of data

In service management home page clicking the "Quickly publish services ", selecting data sources as working space, and then getting into the next step.



1.4.2 Configuring data

Choosing the working space where the data to be used locates, if users select the file type of working space, users can choose the remote working space or the working space on the server, as shown in the figure below:

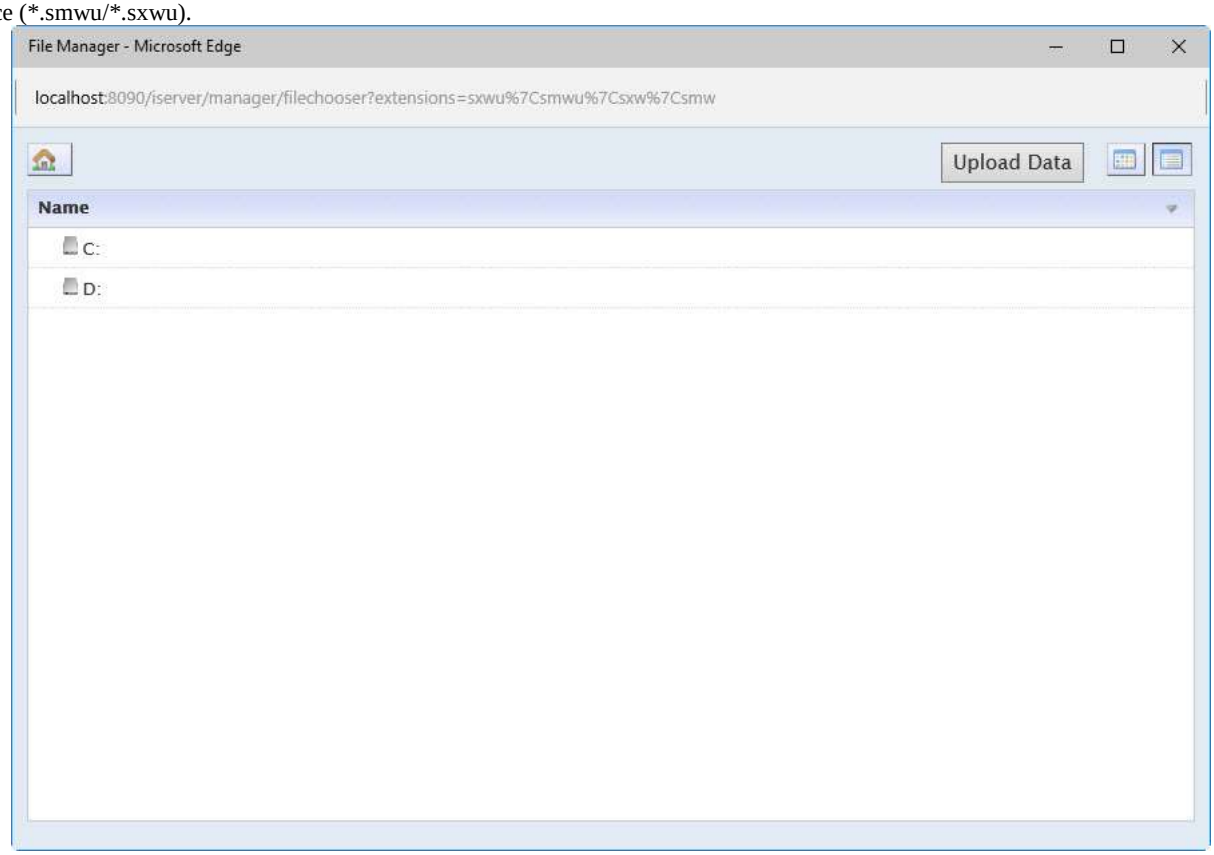


When the service is not at local or uses IE9?IE10?Chrome?Safari browser (Due to the security control of the browser, SuperMap iServer can not get the exact path to publish working space),“ local browsing ” button can not be used, please use “ remotely viewing ”.

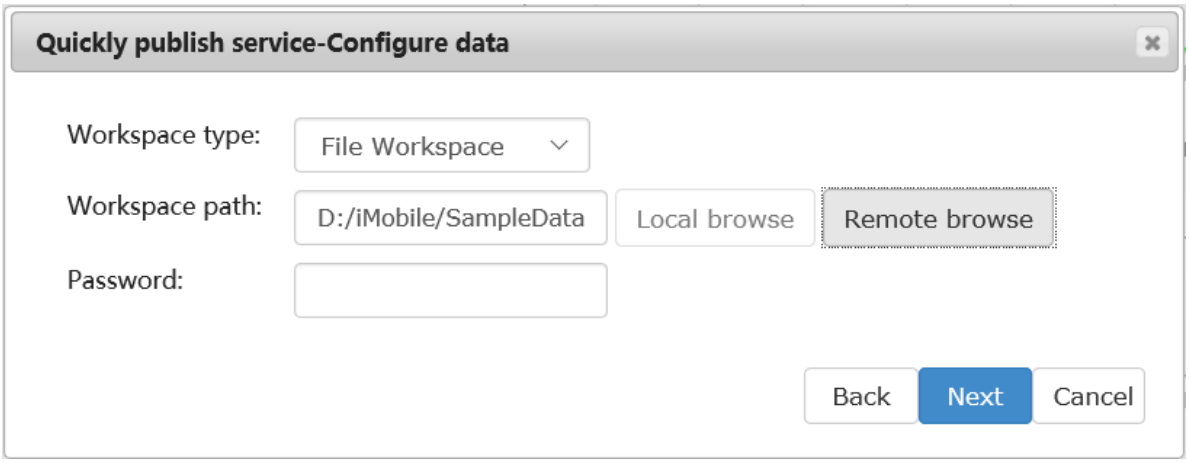
Using " remotely viewing ",SuperMap iServer supports publishing the working space on the remote server, also supports

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

uploading data, as shown in the figure below. Please firstly select target directory then click “ uploading data ” button, and the local data can be uploaded to server and automatically unzipped. Currently supporting .zip zip package and SuperMap UGC 6.x working space (*.smwu/*.sxwu).



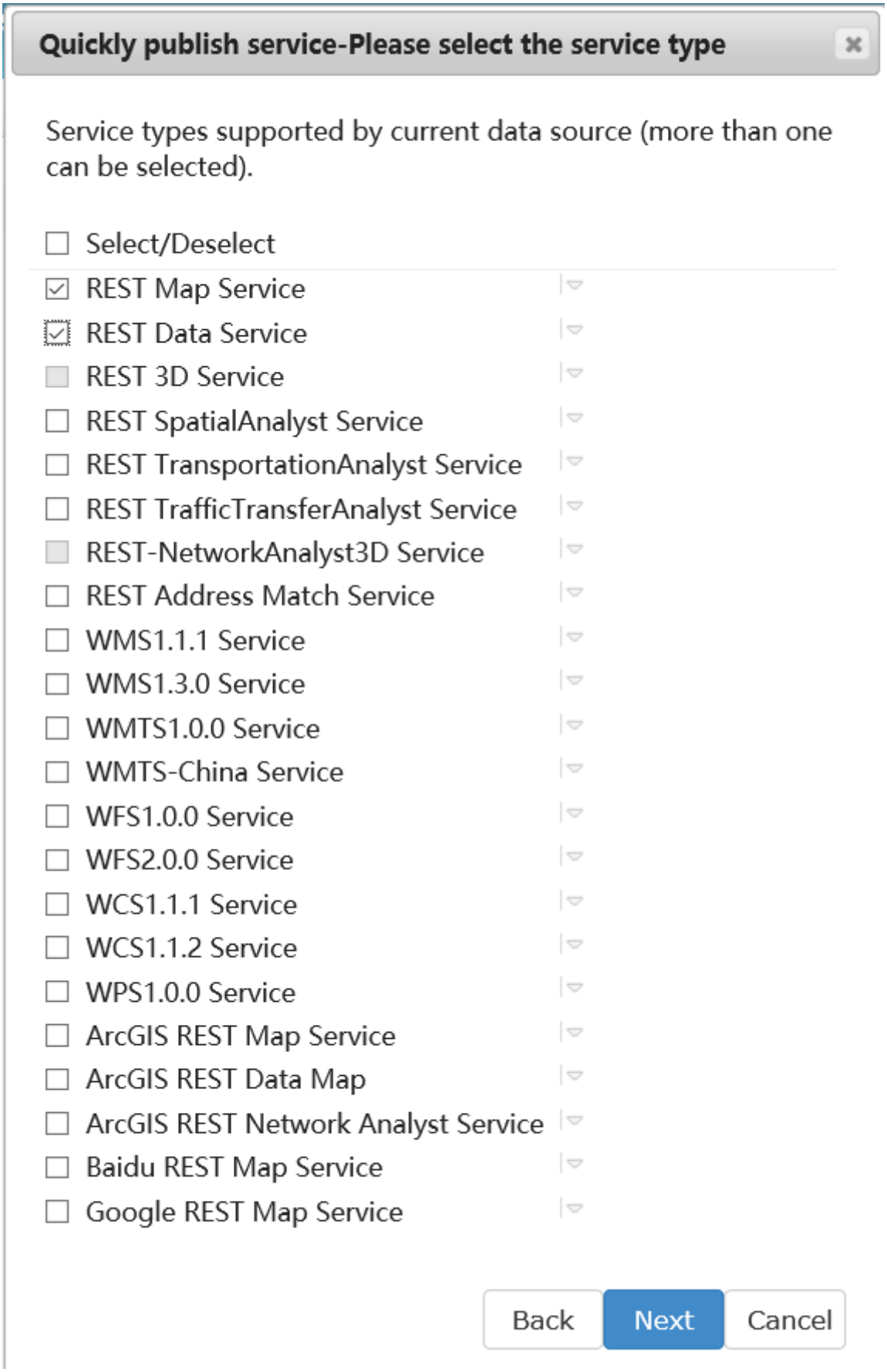
After selecting working space file, if the working space is encrypted and users need to input working space password, if not encrypted, directly clicking on the “ next ” button.



1.4.3 Selecting service type

That is to select service interface type, due to the interaction of the mobile terminal and server-side, uploading and downloading data, querying data are respectively based on the iServer Rest data service and iServer Rest map service. So here we select REST- map service and REST- data service, clicking on the “ next ” button, to get into the next step.

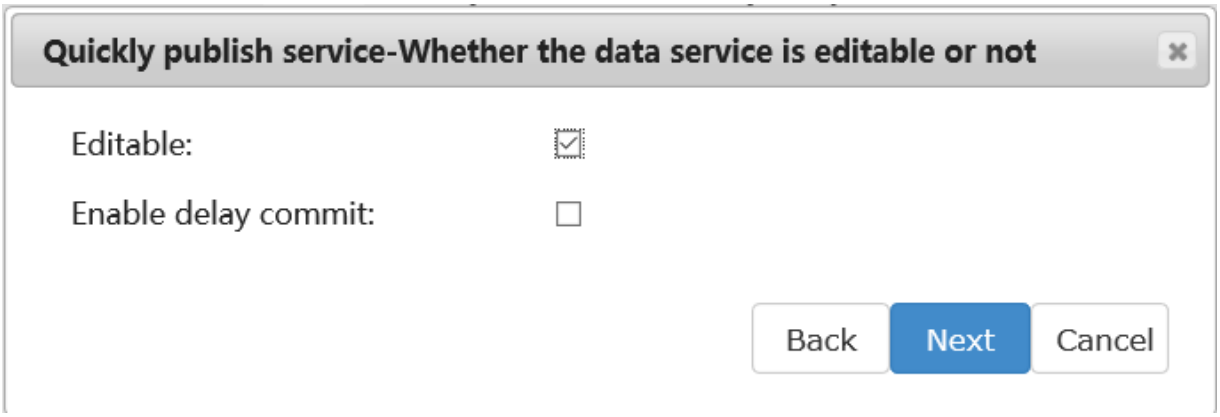
Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com



Based on the requirements of data uploading and downloading service, we need to configure data service to be editable, then click

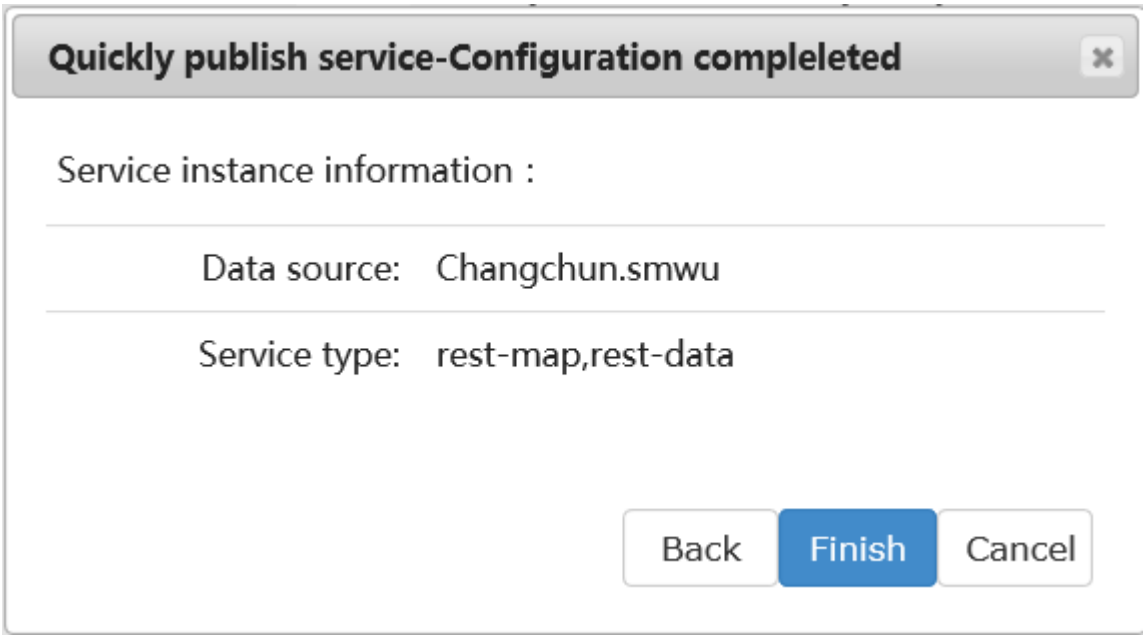
Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

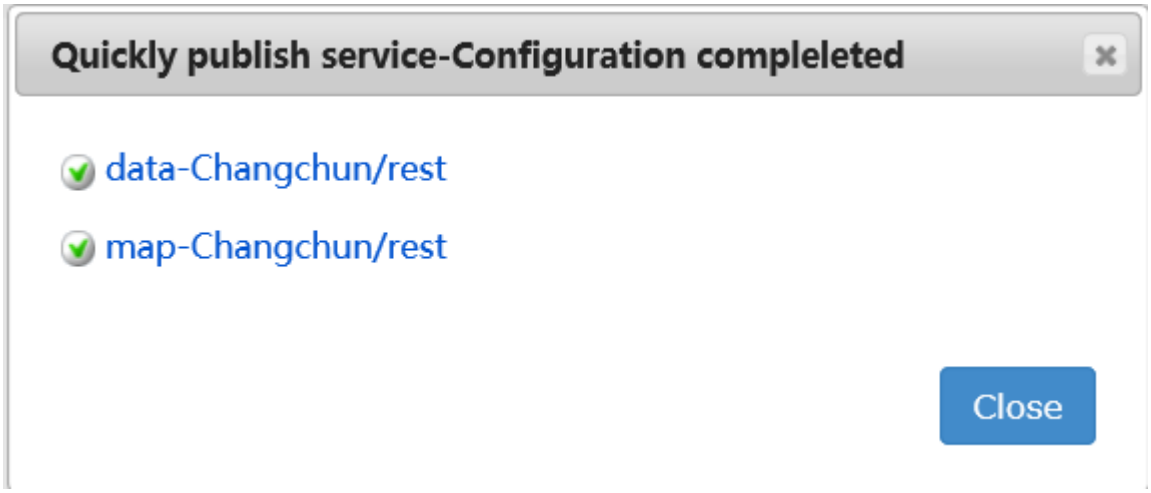
on the “ next ” button, to get into the next step.



1.4.4 Configuring completely

After configuring completely, it will pop up "configuring completely" dialog box, as shown in the figure below: Clicking on the “ complete ”,to complete to create a service instance. In which, the pop-up dialog box after clicking the “ complete ” will give the hyperlinks of service access address.





Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

At this point, the data service and map service required by the interaction of the mobile terminal and server-side are built completely.

2 The interaction between the mobile terminal and server-side implementing to upload and download data

The data uploading and downloading functions of iMobile components products are based on SuperMap iServer Rest data service.

Based on the Changchun data service, local data source World.udb built in the first chapter, this section will elaborate the detailed process of uploading and downloading data, by using iMobile to realize the interaction of mobile terminal and SuperMap iServer server-side.

2.1 First step: Preparing local data source

```
private boolean openDatasource(){
    m_workspace = new Workspace();
    DatasourceConnectionInfo connectionInfo = new
     private final String sdcard = android.os.Environment.getExternalStorageDirectory().
     getAbsolutePath().toString();
      //Local data path
    private final String filepath = sdcard + "/SampleData/GeometryInfo/World.udb";
        connectionInfo.setEngineType(EngineType.UDB);
        connectionInfo.setServer(filepath);
        // Getting data source, by open  get two ways to get data resource.
        //m_datasource = m_workspace.getDatasources().open(connectionInfo);
        m_datasource = m_workspace.getDatasources().get("World.udb");

        if(m_datasource != null)
            return true;

        return false;
       }
```

2.2 Second step: Downing data from server-side

When there is no local data, users can download the required data set from sever-side to local data source. When downloading, the synchronous attribute data set named "downloaded data set name" +"_Table" can be created in local and server-side respectively, used to record the modifications of downloaded data sets in local.

Implementation method of downloading data

```
public void downloadService(){
    if(m_datasource == null)
        return ;
    final ProgressDialog progress = new ProgressDialog(MainFrame.this);
    mUrl="http://localhost:8090/iserver";
     mUrlDatasets ="http://localhost:8090/iserver/services/data-Changchun/rest/data
    /datasources/changchun/datasets";
     datasetName = "School";
    String urlServer  = mUrl;
     String urlDataset = mUrlDatasets + datasetName;
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

```
    // Constructing DataDownloadService objects
     DataDownloadService downloadService = new DataDownloadService(urlServer);
    downloadService.setResponseCallback(new ResponseCallback(){
     @Override
      //The response callback function, is to call back when failing to request to the server
     public void requestFailed(String errorMsg) {
        Log.e("Download", errorMsg);
        progress.dismiss();
        Toast toast = Toast.makeText(m_Activity, "Download failed", Toast.LENGTH_SHORT);
        toast.setGravity(Gravity.CENTER_VERTICAL|Gravity.CENTER_HORIZONTAL, 0, 0);
        toast.show();
    }
     @Override
     //The response callback function, is to call back when succeeding requesting to the server
     public void requestSuccess() {
        Log.i("Download", "requestSuccess");
        }
    @Override
     //The response callback function, called at the end of the data service
     public void dataServiceFinished(String arg0) {
     Toast toast = Toast.makeText(m_Activity, " Download successfully ", Toast.LENGTH_SHORT);
     toast.setGravity(Gravity.CENTER_VERTICAL|Gravity.CENTER_HORIZONTAL, 0, 0);
        toast.show();
        progress.dismiss();
        m_mapControl.getMap().refresh();
    }
});
     // Showing the progress bar executed by downloaded service , destroyed inside the callback
     progress.setMessage(" Downloading ...");
     progress.show();
     //Executing the downloading operations
     downloadService.downloadDataset(urlDataset,m_datasource);

}
```

2.3 Third step: Updating data from server-side to local data sets

When the local has downloaded required data sets, but the server-side data is modified (i.e., the server-side data version is higher than the local data version), the modification of server-side data can be updated to the local data set by the way of updating data. Note, when updating, the server-side and the local both exist the data sets of this name and corresponding synchronous attribute data sets.

In the first step we have downloaded the vector data sets School of Changchun data service to local, here we suppose after we download data, the server-side data set School is modified, then we need to update the modification of server-side data to the local by the way of updating data.

Implementation method of updating data

```
public void updateService(){
    if(m_datasource == null)
            return;
    final ProgressDialog progress = new ProgressDialog(MainFrame.this);
     mUrl="http://localhost:8090/iserver";
     mUrlDatasets ="http://localhost:8090/iserver/services/data-Changchun/rest/data
    /datasources/changchun/datasets";
     datasetName = "School";
    String urlServer  = mUrl;
    String urlDataset =mUrlDatasets + datasetName;
    // Constructing DataDownloadService objects
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

```
DataDownloadService downloadService = new DataDownloadService(urlServer);
downloadService.setResponseCallback(new ResponseCallback(){

@Override
//The response callback function, is to call back when failing to request to the server
public void requestFailed(String errorMsg) {
    Log.e("Update: ", errorMsg);
    progress.dismiss();
    Toast toast = Toast.makeText(m_Activity, " Update failed ", Toast.LENGTH_SHORT);
    toast.setGravity(Gravity.CENTER_VERTICAL|Gravity.CENTER_HORIZONTAL, 0, 0);
    toast.show();
    }
@Override
//The response callback function, is to call back when succeeding requesting to the server
public void requestSuccess() {
        System.out.println(" Update completely ");
    }
@Override
//The response callback function, called at the end of the data service
public void dataServiceFinished(String arg0) {
    Toast toast = Toast.makeText(m_Activity, " Update successfully ", Toast.LENGTH_SHORT);
    toast.setGravity(Gravity.CENTER_VERTICAL|Gravity.CENTER_HORIZONTAL, 0, 0);
    toast.show();
    progress.dismiss();
    m_mapControl.getMap().refresh();
        }
    });
     // Showing the progress bar executed by updated service , destroyed inside the callback
    progress.setMessage(" Updating ...");
    progress.show();
    //Executing the updating operations
    downloadService.updateDataset(urlDataset, (DatasetVector)m_datasource.getDatas
     sets().get(datasetName));
}
```

2.4 Fourth step: Modifying and updating / the data downloaded in local

Updating the data to the local by using the method provided in the second step, the local data can be modified according to the business needs.

Note: When modifying data, the data can not be modified by the adding, modifying, deleting data interface provided by DataUploadService class, because the 14 methods the class provides are directly operated on the server-side the data, do not support the version management function.

This example modified the vector data set , named School downloaded in local in the second step:

Implementation method of modifying data

```
protected void modifiedDataset(int type){
    // Getting related data sets
    DatasetVector dataset = (DatasetVector)m_datasource.getDatasets().get(datasetName);
        if(dataset == null)
            return;
    // Getting records to edit
    Recordset recordset = dataset.getRecordset(false, CursorType.DYNAMIC);
        switch(type){
        // Modifying the filed value of the filed whose filed name is ”SMUSERID” as 20
        case 1:
            if(!recordset.isEmpty()){
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

```
                while(!recordset.isEOF()){
                    recordset.edit();
                    recordset.setFieldValue("SMUSERID", 20);
                    recordset.update();
                    recordset.moveNext();
                }
            }
            recordset.update();
            recordset.dispose();
            break;
        // Adding 2 point geometric objects in the record sets
        case 2:
            if(!recordset.isEmpty()){
                GeoPoint point1 = new GeoPoint (7056.451, 34690.675);
                recordset.addNew(point1);
                recordset.update();
                recordset.moveNext();
                GeoPoint point2  = new GeoPoint (6254.780, 54220.603);
                recordset.addNew(point2);
                recordset.update();
            }

        default:
                break;
        }

    }
```

2.5 **Fifth step: Submitting data to server-side**

After completing to modifying local data as business needs, the modified data can be submitted to server-side Note two points when submitting data: 1)when updating, the server-side and the local both exist the data sets of this name and corresponding synchronous attribute data sets. 2)when updating, the local data version can not be higher than the server-side data version (i.e., the local Max[SmUserID] is not lager than Max[SMID] of server).
In this example, showing how to submit the School data set modified in the third step to server-side:

```
public void uploadService(){
    final ProgressDialog progress = new ProgressDialog(MainFrame.this);
    // Calling methods in the third step, modifying data
    modifiedDataset(2);
    DatasetVector dataset = (DatasetVector)m_datasource.getDatasets().get(datasetName);
     mUrl="http://localhost:8090/iserver";
     mUrlDatasets ="http://localhost:8090/iserver/services/data-Changchun/rest/data
    /datasources/changchun/datasets";
     datasetName = "School";
     String urlServer  = mUrl;
    String urlDataset =mUrlDatasets + datasetName;
    // Constructing DataUploadService objects
    DataUploadService uploadService = newDataUploadService(urlServer);
    uploadService.setResponseCallback(new ResponseCallback(){
    @Override
   //The response callback function, is to call back when failing to request to the server
    public void requestFailed(String errorMsg) {
        Log.e("commit: ", errorMsg);
        progress.dismiss();
        Toast toast = Toast.makeText(m_Activity, " Submit failed ", Toast.LENGTH_SHORT);
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

```
        toast.setGravity(Gravity.CENTER_VERTICAL|Gravity.CENTER_HORIZONTAL, 0, 0);
        toast.show();
    }
    @Override
    //The response callback function, is to call back when succeeding requesting to the server
    public void requestSuccess() {
        Log.i("Commit", "requestSuccess");
    }
    @Override
    //The response callback function, called at the end of the data service
    public void dataServiceFinished(String arg0) {
        Toast toast = Toast.makeText(m_Activity, " Submit successfully ", Toast.LENGTH_SHORT);
        toast.setGravity(Gravity.CENTER_VERTICAL|Gravity.CENTER_HORIZONTAL, 0, 0);
        toast.show();
        progress.dismiss();
        m_mapControl.getMap().refresh();
    }
});
        // Showing the progress bar executed by submitted service , destroyed inside the callback
        progress.setMessage(" Submitting ...");
        progress.show();
        //Executing submitting operation
        uploadService.commitDataset(urlDataset, dataset);
    }
```

3 The interaction of the mobile terminal and server-side implementing to query server-side data

The data querying function of iMobile components products is based on SuperMap iServer Rest data service. The query results of the server need to return by the way of Featureset elements set. Users need to set the response callback function, through the callback function, getting the operation results and server response, etc.

Based on the Changchun map service built in the first chapter, this section will elaborate the detailed process of querying data, by using iMobile to realize the interaction of mobile terminal and SuperMap iServer server-side.

3.1 First step: Visiting map service, and adding it to map window in the way of layer

```
private boolean openMap(){
    m_woWorkspace = new Workspace();
    // Associating the map displaying controls with work space
    m_mapView = (MapView)findViewById(R.id.mapview);
    m_mapControl = m_mapView.getMapControl();
     m_mapControl.getMap().setWorkspace(m_woWorkspace);
     DatasourceConnectionInfo dsInfo = new DatasourceConnectionInfo();
   dsInfo.setServer("http://localhost:8090/iserver/services/map-Changchun/rest/maps/ city map of Changchun ");
    dsInfo.setEngineType(EngineType.Rest);
    dsInfo.setAlias("ChinaRest");
    Datasource ds = m_woWorkspace.getDatasources().open(dsInfo);
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

```
    if(ds != null){
         m_mapControl.getMap().getLayers().add(ds.getDatasets().get(0), true);
         m_mapControl.getMap().refresh();
         return true;
    }
        Log.e(this.getClass().getName(), " open data source failed ");
          return true;

        }
```

3.2 Second step: Querying server-side data

```
private void Query() {
    m_mapView.removeAllCallOut();

    final ProgressDialog progress = new ProgressDialog(MainFrame.this);
    // Constructing querying service object Queryservice
    m_strServer ="http://localhost:8090/iserver";
    QueryService service = new QueryService(m_strServer);
    // Constructing querying service object ServiceQueryParameter
    ServiceQueryParameter parameter = new ServiceQueryParameter();
    // Setting querying parameter
    parameter.setQueryMapName(m_edtMapName.getText().toString());
    parameter.setQueryServiceName(m_edtServerName.getText().toString());
    parameter.setQueryLayerName(m_edtLayerName.getText().toString());
    parameter.setExpectRecordCount(100);
    parameter.setQueryRecordStart(0);
    parameter.setQueryOption(QueryOption.GEOMETRY);
    parameter.setAttributeFilter(m_edtSql.getText().toString());

    service.setResponseCallback(new ResponseCallback() {
    @Override
    //The response callback function, is to call back when failing to request to the server
    public void requestSuccess() {
        // Destroying the progress showing bar
        progress.dismiss();
        Toast.makeText(MainFrame.this, " query successfully ", Toast.LENGTH_LONG).show();
    }
    @Override
    //The response callback function, is to call back when failing to request to the server
    public void requestFailed(String arg0) {
        // Destroying the progress showing bar
        progress.dismiss();
        Toast.makeText(MainFrame.this, " query failed ", Toast.LENGTH_LONG).show();
        System.out.println(" error information " + arg0);
    }
    @Override
    //The response callback function, is to call back when receiving response results of the server
    public void receiveResponse(FeatureSet arg0) {
        if (arg0 instanceof FeatureSet) {
            FeatureSet featureSet = (FeatureSet)arg0;
            int nCount = 0;
            featureSet.moveFirst();
        while (!featureSet.isEOF()) {
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

```
            Geometry geo = featureSet.getGeometry();
             if (geo == null) {
                featureSet.moveNext();
                continue;
            }
            nCount++;
            Point2D pt = featureSet.getGeometry().getInnerPoint();
            LayoutInflater lfCallOut = getLayoutInflater();
            View calloutLayout = lfCallOut.inflate(R.layout.callout, null);
            CallOut callout = new CallOut(MainFrame.this);
             // Setting display contents
             callout.setContentView(calloutLayout);
            // Setting custom background photos
             callout.setCustomize(true);
             // Setting displaying location
            callout.setLocation(pt.getX(), pt.getY());
            m_mapView.addCallout(callout);
            featureSet.moveNext();
        }
        System.out.println("count is " + nCount);
        System.out.println("featureSet count is " + featureSet.getFeatureCount());
        }
}
        // Displaying the progress bar of querying service, destroyed inside the callback
        progress.setMessage(" querying service ...");
        progress.show();
          // Executing query operation
        service.query(parameter, QueryMode.SqlQuery);
```

4 Online map service access

4.1 Docking Baidu map service

```
    DatasourceConnectionInfo info = newDatasourceConnectionInfo();
    // Setting data source alias
    info.setAlias("BaiDu1");
    // Setting engine types
    info.setEngineType(EngineType.BaiDu);
    //Setting the map service address（when docking Baidu map service, it can be opened without setting the item, if the service address changes, the changed Baidu map service address can be set here）
    String url = "https://map.baidu.com";
    info.setServer(url);
    // Opening data source
    Datasource datasource = m_workspace.getDatasources().open(info);
    // Adding data sets to map window
    m_mapControl = m_mapView.getMapControl();
    m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(0),true);
```

4.2 Docking Bing maps service

4.2.1 Docking Chinese Bing maps service

```
    DatasourceConnectionInfo info = newDatasourceConnectionInfo();
    // Setting data source alias
    info.setAlias("BingMaps1");
    // Setting engine types
    info.setEngineType(EngineType.BingMaps);
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

```
//Setting the map service address (when docking Bing maps service, it can be opened without setting the item, if the service address changes, the changed Bing maps service address can be set here)
String url = "https://www.microsoft.com/maps";
info.setServer(url);
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(0),true);
// Setting the display scale of current map
m_mapControl.getMap().setScale(1.703471946182E-8);
```

4.2.2 **Docking English Bing maps service**

```
DatasourceConnectionInfo info = newDatasourceConnectionInfo();
// Setting data source alias
info.setAlias("BingMap");
// Setting Key value
info.setPassword("bing map service key value ");
// Setting engine types
info.setEngineType(EngineType.BingMaps);
// Setting the map service address
String path = "https://www.microsoft.com/maps";
infoBing.setServer(path);
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(2/*1*/),true);
// Setting the display scale of current map
m_mapControl.getMap().setScale(1.703471946182E-8);
```

4.3 **Docking Google Maps service**

```
DatasourceConnectionInfo info = newDatasourceConnectionInfo();
// Setting data source alias
info.setAlias("GoogleMapRoad");
// Setting engine types
info.setEngineType(EngineType. Google Maps );
//Setting the map service address (when docking Google Maps service, it can be opened without setting the item, if the service address changes, the changed Google Maps service address can be set here)
String url ="http://www.google.cn/maps";
info.setServer(url);
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get("roadmap"),true);
```

4.4 **Docking *OpenStreetMap***

```
DatasourceConnectionInfo info = newDatasourceConnectionInfo();
// Setting data source alias
info.setAlias("OpenStreetMap2");
// Setting engine types
info.setEngineType(EngineType.OpenStreetMaps);
//Setting the map service address ( when docking OpenStreetMap maps service, it can be opened without setting the item
// if the service address changes, the changed OpenStreetMap service address can be set here )
String url = "https://openstreetmap.org";
info.setServer(url);
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

```
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(0),true);
```

4.5 Docking Tianditu service

```
DatasourceConnectionInfo info = newDatasourceConnectionInfo();
// Setting data source alias
info.setAlias("TianDiTu1");
// Setting engine types
info.setEngineType(EngineType.OGC);
// Setting drive name
info.setDriver("WMTS");
//Setting the map service address
String url = "http://t0.tianditu.com/vec_c/wmts";
info.setServer(url);
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(0),true);
```

4.6 Docking Supermap cloud map service

```
DatasourceConnectionInfo info = newDatasourceConnectionInfo();
// Setting data source alias
info.setAlias("SuperMapCloud1");
// Setting engine types
info.setEngineType(EngineType.SuperMapCloud);
//Setting the map service address ( is compulsory )
String url = " http://supermapcloud.com";
info.setServer(url);
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(0),true);
```

4.3 Docking Rest map service

```
DatasourceConnectionInfo info = new DatasourceConnectionInfo();
// Setting data source alias
info.setAlias("Rest1");
// Setting engine types
info.setEngineType(EngineType.Rest);
// Map service address ( Here takes locally built example service as an example: )
// Note:
//1. The Internet connected by mobile phone and the service address need to be in the same network segment
//2. The local built example service, service address is unavailable localhost , it needs to input ip address
String url = "http:// 192.168.12.12:8090/iserver/services/map-world/rest/maps/World";
//Setting the map service address ( is compulsory )
info.setServer(url);
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

```
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(0),true);
```

4.8 Docking OGC service

4.8.1 Docking WMS map service

```
// Open by default mode WMS service data:
DatasourceConnectionInfo info = new DatasourceConnectionInfo();
//Setting the map service address ( is compulsory , Here takes locally built example service as an example: )
info.setServer("http://192.168.12.12:8090/iserver/services/map-world/wms130/World ");
// Setting engine types
info.setEngineType(EngineType.OGC);
// Setting drive name
info.setDriver("WMS");
// Setting data source alias
info.setAlias("openWMS");
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(0),true);
```

```
// Open by external parameter mode WMS service data:
DatasourceConnectionInfo info = new DatasourceConnectionInfo();
//Setting the map service address ( )
info.setServer("http://192.168.12.12:8090/iserver/services/map-world/wms130/World");
// Setting engine types
info.setEngineType(EngineType.OGC);
// Setting drive name
info.setDriver("WMS");
// Setting WMS service version number
info.setWebVersion("1.3.0");
// Setting WMS service image format
info.setWebFormat("image/png");
// Setting WMS service visible layers
info.setWebVisibleLayers("0.10,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0.0");
// Setting Web service data source coordinate reference system
info.setWebCoordinate("EPSG:3857");
Rectangle2D rect = new Rectangle2D(-1.003750834278E7, -1.003750834279E7, 1.003750834278E7,
1.003750834279E7);
// Setting WMS service map scope
info.setWebBBox(rect);
// Setting WMS service extended parameters
info.setWebExtendParam("");
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(0),true);
```

4.8.2 Docking WMTS map service

```
DatasourceConnectionInfo info = new DatasourceConnectionInfo();
// Setting data source alias
info.setAlias("WMTS1");
// Setting engine types
info.setEngineType(EngineType.OGC);
// Setting drive name
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com

```
info.setDriver("WMTS");
//Setting the map service address ( is compulsory ， Here takes locally built example service as an example: )
String url = "http://192.168.12.12:8090/iserver/services/map-world/wmts-china";
info.setServer(url);
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(0),true);
```

4.8.3 Docking WFS map service

```
DatasourceConnectionInfo info = new DatasourceConnectionInfo();
// Setting data source alias
info.setAlias("WFS1");
// Setting engine types
info.setEngineType(EngineType.OGC);
// Setting drive name
info.setDriver("WFS");
//Setting the map service address ( is compulsory , Here takes locally built example service as an example: )
String url = " http://192.168.12.12:8090/iserver/services/data-world/wfs100/gb18030";
info.setServer(url);
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(datasource.getDatasets().get(0),true);
```

4.8.4 Docking WCS map service

```
DatasourceConnectionInfo info = new DatasourceConnectionInfo();
//Setting the map service address ( is compulsory ， Here takes locally built example service as an example: )
String url = "http://support.supermap.com.cn:8090/iserver/services/data-world/wcs111";
info.setServer(url);
// Setting drive name
info.setDriver("WCS");
// Setting engine types
info.setEngineType(EngineType.OGC);
// Setting data source alias
info.setAlias("openWCS");
// Opening data source
Datasource datasource = m_workspace.getDatasources().open(info);
// Adding data sets to map window
m_mapControl = m_mapView.getMapControl();
m_mapControl.getMap().getLayers().add(m_datasource.getDatasets().get(0),true);
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.comWebsite: www.supermap.com