DNSMANAGER

MAOYU WANG, HAOTIAN ZHANG, SITONG LIU, RUOYAN WU

BRIFE DESCRIBTION

DNSManager is a client that provides user with a rich GUI to manage their domain records

FUNCTIONA LITY

Users often have their domains hosted in different registers such as **godaddy**, **digitalocean**, **cloudflare** and so on. They need to login in on different registers' websites to make changes to their domain records.
DNSManager is such a client that users can just manager all their domains in one place. This client only needs API keys.

FRAMEWORKS

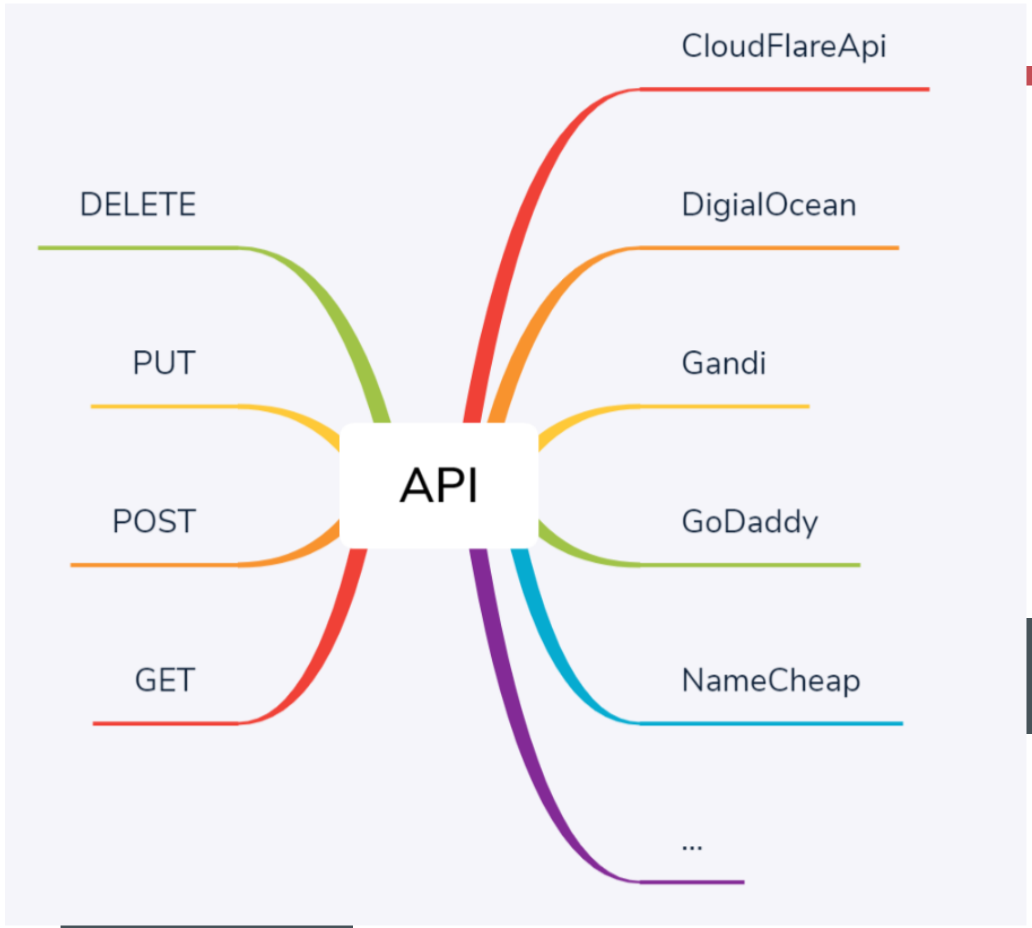
- SWING GUI
  FASTJSON
  SQLITE
  JAVA JDBC
  RESTFUL API

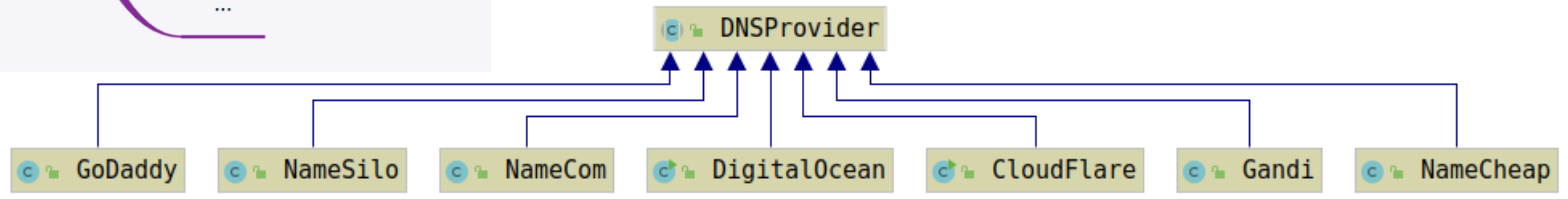
SUPPORTED WEBSITES

CloudFlare
DigitalOcean
Godaddy
NameCheap
NameSilo
Name.Com
Gandi

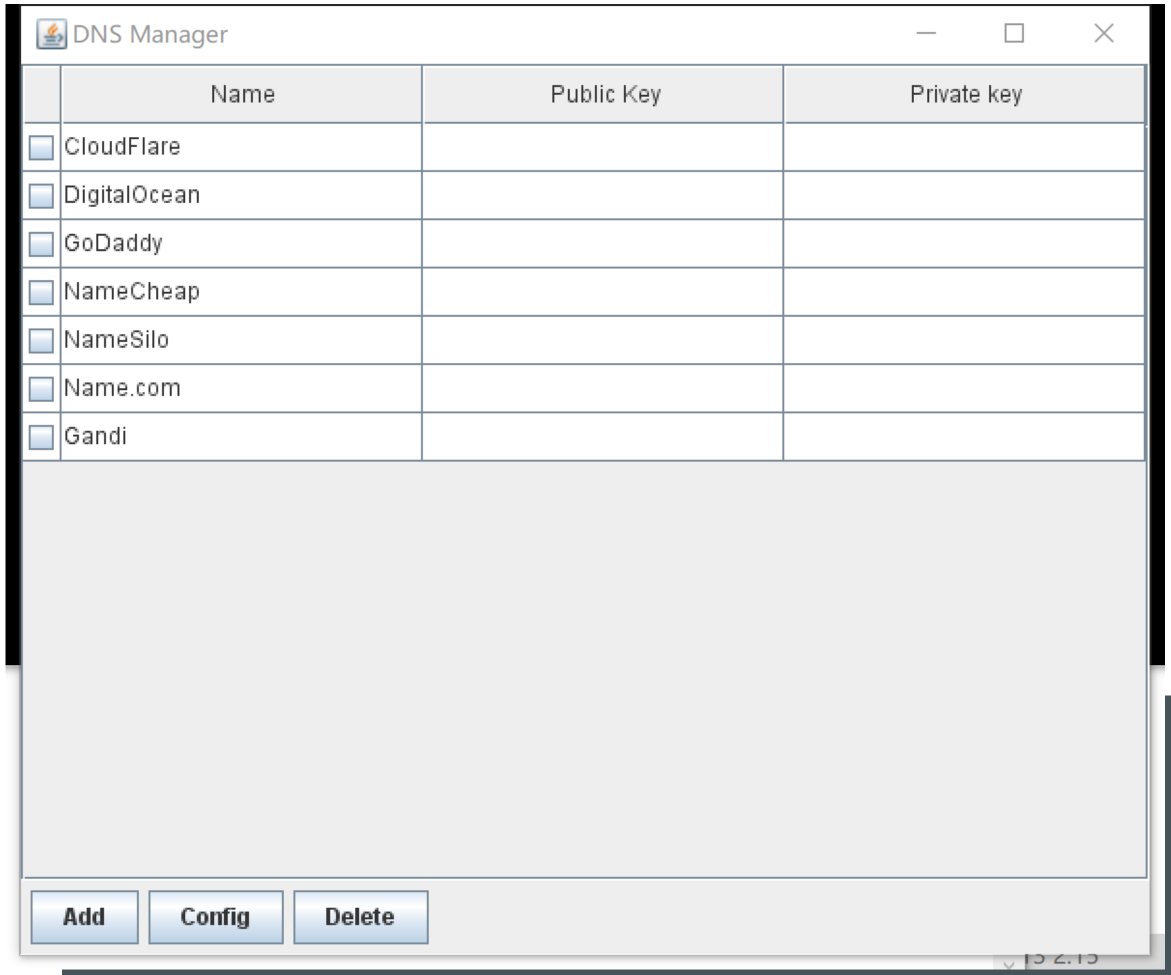
A
CNAME
AAAA
TXT

SUPPORTED RECORD TYPE


API
DELETE
PUT
POST
GET
CloudFlareApi
DigialOcean
Gandi
GoDaddy
NameCheap
...

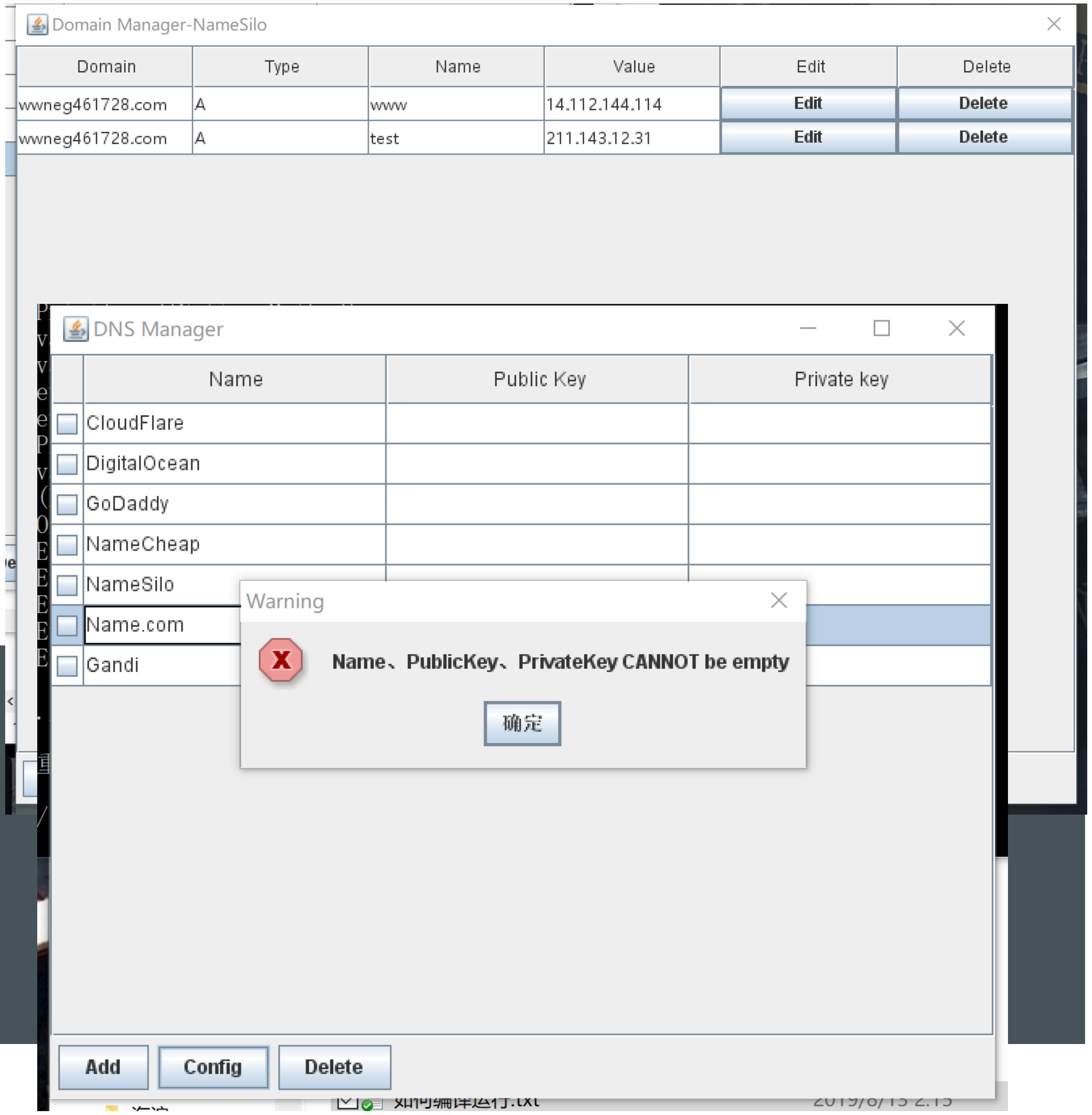

DNSProvider
GoDaddy
NameSilo
NameCom
DigitalOcean
CloudFlare
Gandi
NameCheap

DIAGRAM



DNS Manager

	Name	Public Key	Private key
☐	CloudFlare		
☐	DigitalOcean		
☐	GoDaddy		
☐	NameCheap		
☐	NameSilo		
☐	Name.com		
☐	Gandi		



Domain Manager-NameSilo

Domain	Type	Name	Value	Edit	Delete
wwneg461728.com	A	www	14.112.144.114	Edit	Delete
wwneg461728.com	A	test	211.143.12.31	Edit	Delete

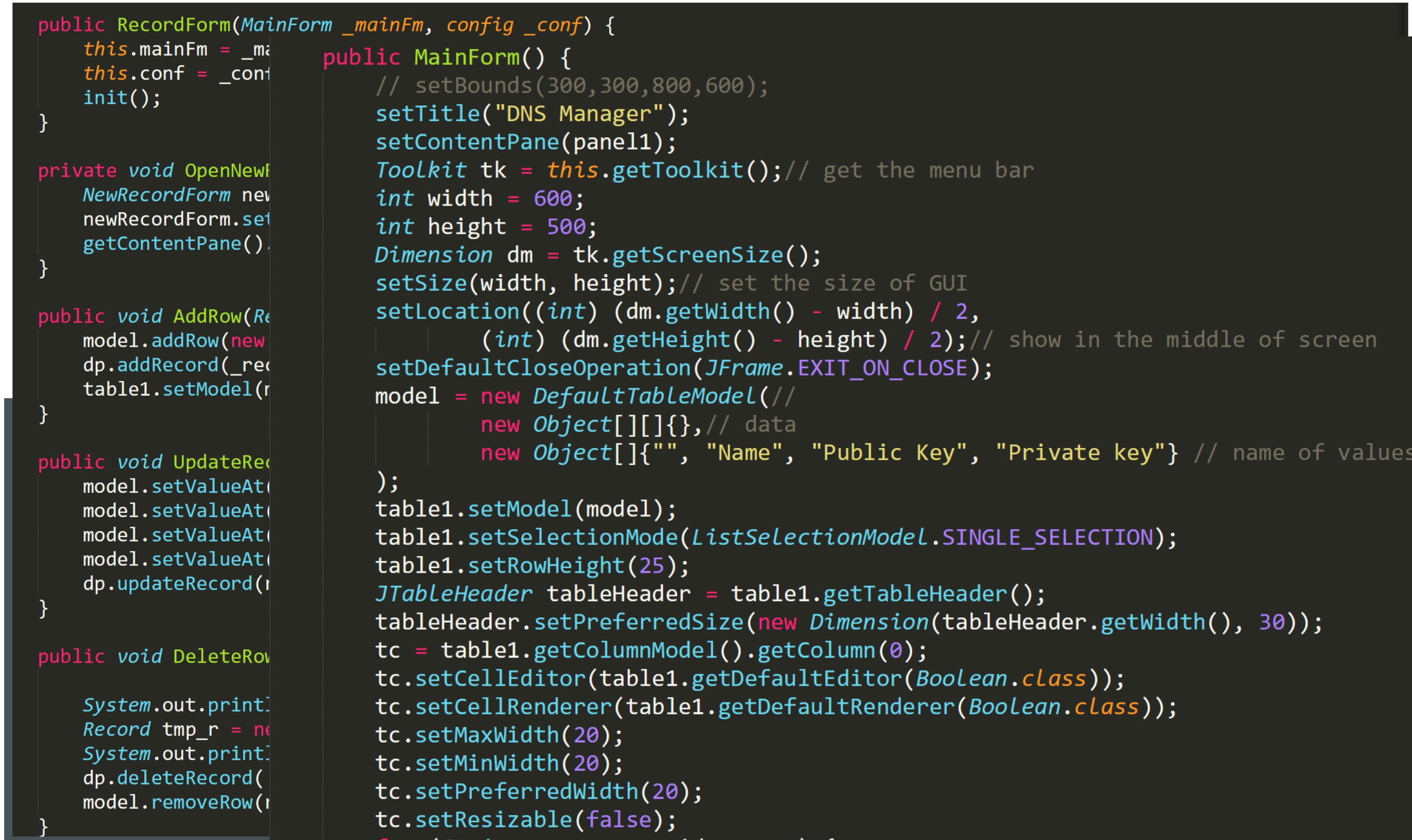
DNS Manager

	Name	Public Key	Private key
☐	CloudFlare		
☐	DigitalOcean		
☐	GoDaddy		
☐	NameCheap		
☐	NameSilo		
☐	Name.com		
☐	Gandi		

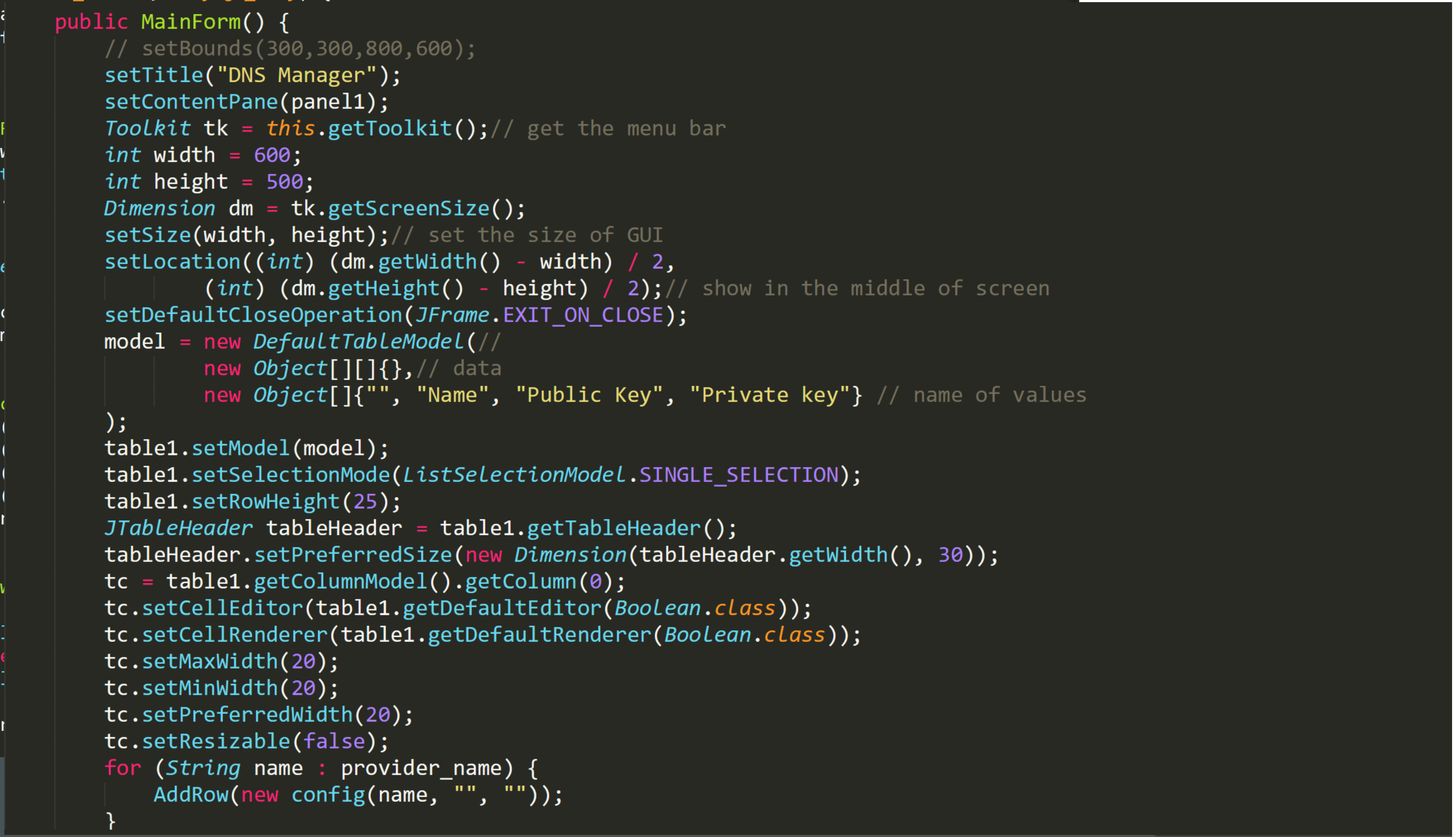
Warning

Name、PublicKey、PrivateKey CANNOT be empty

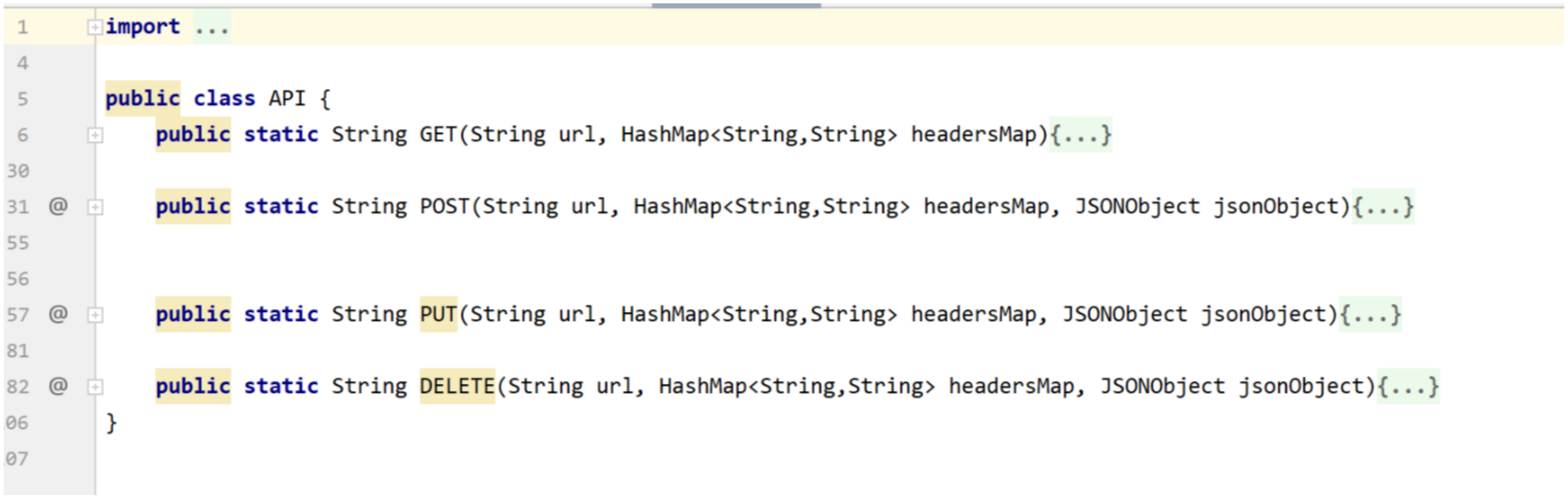
FRONTEND



```
public RecordForm(MainForm _mainFm, config _conf) {
    this.mainFm = _ma
    this.conf = _conf
    init();
}

private void OpenNewR
    NewRecordForm new
    newRecordForm.set
    getContentPane().
}

public void AddRow(Re
    model.addRow(new
    dp.addRecord(_rec
    table1.setModel(r
}

public void UpdateRec
    model.setValueAt(
    model.setValueAt(
    model.setValueAt(
    model.setValueAt(
    dp.updateRecord(r
}

public void DeleteRow

    System.out.printl
    Record tmp_r = ne
    System.out.printl
    dp.deleteRecord(
    model.removeRow(r
}
```

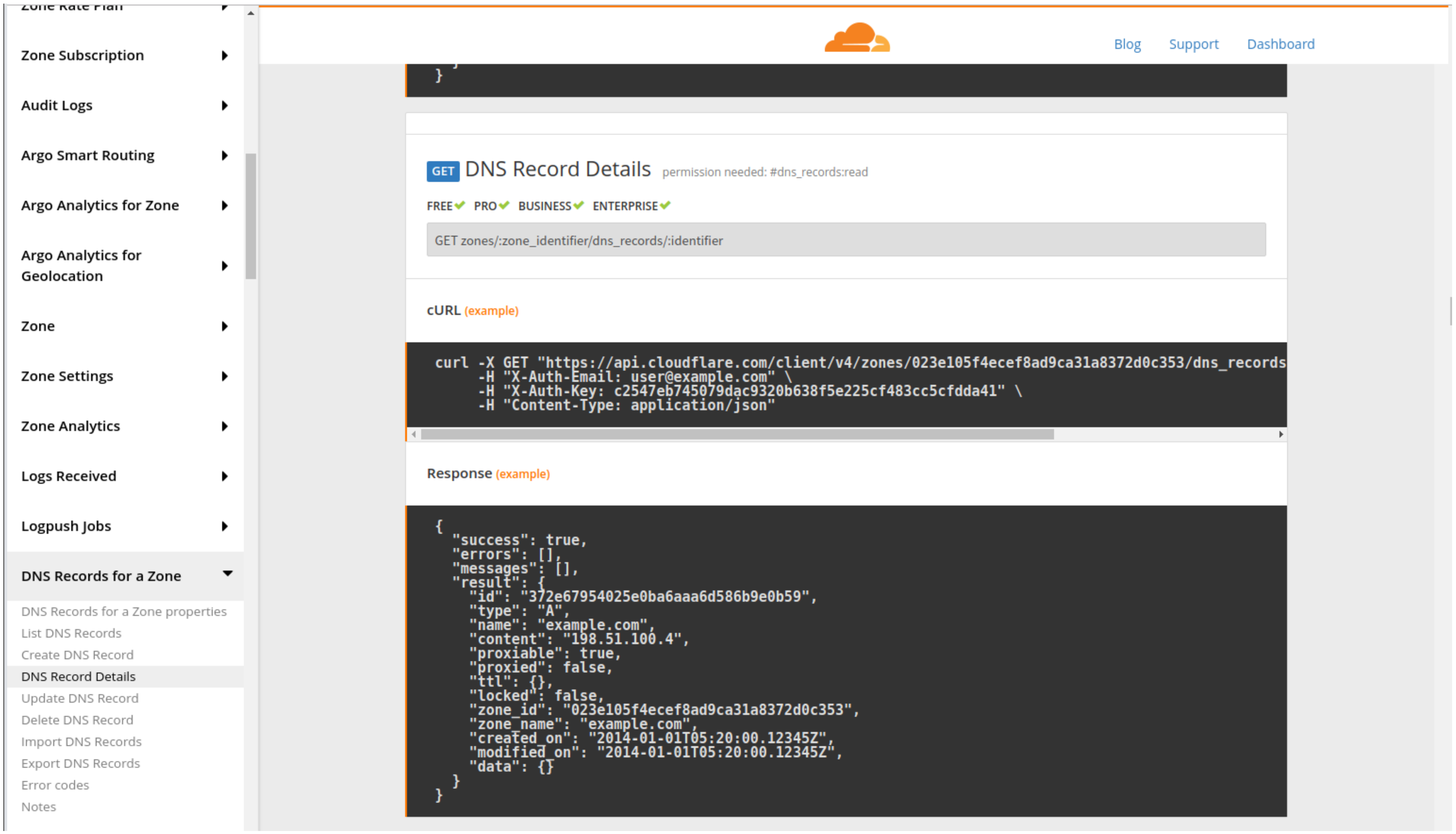


```
public MainForm() {
    // setBounds(300,300,800,600);
    setTitle("DNS Manager");
    setContentPane(panel1);
    Toolkit tk = this.getToolkit();// get the menu bar
    int width = 600;
    int height = 500;
    Dimension dm = tk.getScreenSize();
    setSize(width, height);// set the size of GUI
    setLocation((int) (dm.getWidth() - width) / 2,
            (int) (dm.getHeight() - height) / 2);// show in the middle of screen
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    model = new DefaultTableModel(//
            new Object[][]{},// data
            new Object[]{"", "Name", "Public Key", "Private key"} // name of values
    );
    table1.setModel(model);
    table1.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
    table1.setRowHeight(25);
    JTableHeader tableHeader = table1.getTableHeader();
    tableHeader.setPreferredSize(new Dimension(tableHeader.getWidth(), 30));
    tc = table1.getColumnModel().getColumn(0);
    tc.setCellEditor(table1.getDefaultEditor(Boolean.class));
    tc.setCellRenderer(table1.getDefaultRenderer(Boolean.class));
    tc.setMaxWidth(20);
    tc.setMinWidth(20);
    tc.setPreferredWidth(20);
    tc.setResizable(false);
    for (String name : provider_name) {
        AddRow(new config(name, "", ""));
    }
```



```
import ...

public class API {
    public static String GET(String url, HashMap<String,String> headersMap){...}

@   public static String POST(String url, HashMap<String,String> headersMap, JSONObject jsonObject){...}


@   public static String PUT(String url, HashMap<String,String> headersMap, JSONObject jsonObject){...}

@   public static String DELETE(String url, HashMap<String,String> headersMap, JSONObject jsonObject){...}
}
```

BACKEND



```
  }
```

GET **DNS Record Details** permission needed: #dns_records:read

FREE ✓ PRO ✓ BUSINESS ✓ ENTERPRISE ✓

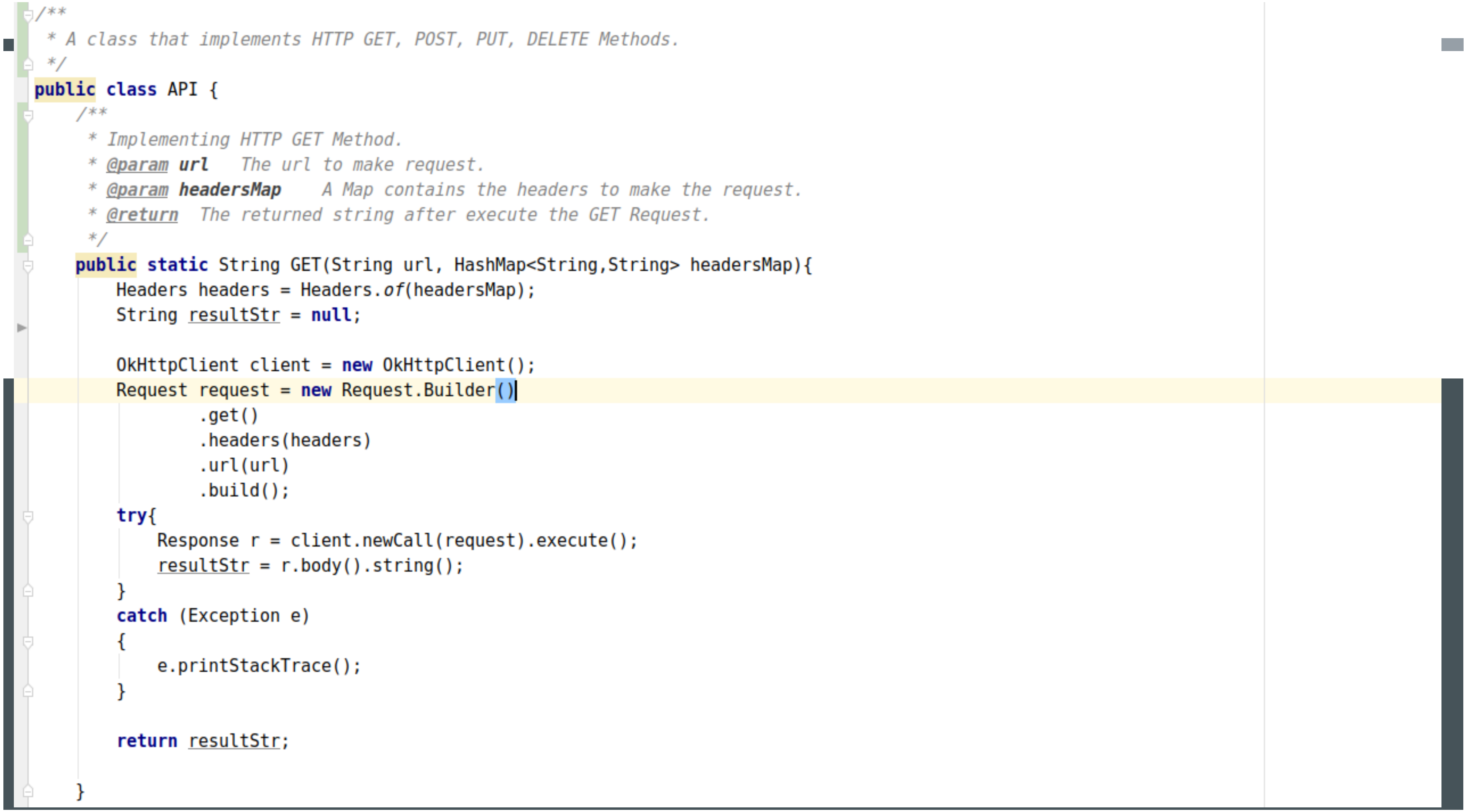
GET zones/:zone_identifier/dns_records/:identifier

cURL (example)

```
curl -X GET "https://api.cloudflare.com/client/v4/zones/023e105f4ecef8ad9ca31a8372d0c353/dns_records
     -H "X-Auth-Email: user@example.com" \
     -H "X-Auth-Key: c2547eb745079dac9320b638f5e225cf483cc5cfdda41" \
     -H "Content-Type: application/json"
```

Response (example)

```
{
  "success": true,
  "errors": [],
  "messages": [],
  "result": {
    "id": "372e67954025e0ba6aaa6d586b9e0b59",
    "type": "A",
    "name": "example.com",
    "content": "198.51.100.4",
    "proxiable": true,
    "proxied": false,
    "ttl": {},
    "locked": false,
    "zone_id": "023e105f4ecef8ad9ca31a8372d0c353",
    "zone_name": "example.com",
    "created_on": "2014-01-01T05:20:00.12345Z",
    "modified_on": "2014-01-01T05:20:00.12345Z",
    "data": {}
  }
}
```



```
/**
 * A class that implements HTTP GET, POST, PUT, DELETE Methods.
 */
public class API {
    /**
     * Implementing HTTP GET Method.
     * @param url   The url to make request.
     * @param headersMap    A Map contains the headers to make the request.
     * @return  The returned string after execute the GET Request.
     */
    public static String GET(String url, HashMap<String,String> headersMap){
        Headers headers = Headers.of(headersMap);
        String resultStr = null;

        OkHttpClient client = new OkHttpClient();
        Request request = new Request.Builder()
                .get()
                .headers(headers)
                .url(url)
                .build();
        try{
            Response r = client.newCall(request).execute();
            resultStr = r.body().string();
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

        return resultStr;

    }
```

DEMO

FURTHER DEVELOPMENT

If we have extra time, we can develop a SSL certificates manager module to help manage SSL certs by using OPENSSL framework and Let’s Encrypt.

Q&A

```
object to mirror
irror_mod.mirror_object
operation == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

selection at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.active
("Selected" + str(modifier
irror_ob.select = 0
bpy.context.selected_ob
ata.objects[one.name].se

int("please select exactly

- OPERATOR CLASSES ----

types.Operator):
X mirror to the selected
ject.mirror_mirror_x"
rror X"
```

THANKS!