

12. json数据接收与解析

服务器端返回的数据用Json格式打包，在显示时要进行解析。

```
{"employeeID":%1,"name":%2,"department":软件,"time":%3}
```

步骤：

1. 在头文件中加入Json相关库，并新建Mat对象存储人脸。

```
#include <QJsonDocument>
#include <QJsonParseError>
#include <QJsonObject>
```

```
// 保存人脸的数据
cv::Mat faceMat;
```

2. 在 void FaceAttendance::timerEvent(QTimerEvent *e) 中检测到图像后保存图像。

```
faceMat = srcImage(rect);
// 2. 通过样式显示
imwrite("./face.jpg",faceMat);
```

3. 接收数据后转换Json

```
void FaceAttendance::recv_data(){
    QByteArray array = msocket.readAll();
    // Json文件解析
    QJsonParseError err;
    QJsonDocument doc = QJsonDocument::fromJson(array,&err);
    if(err.error != QJsonParseError::NoError){
        qDebug()<<"json数据错误";
        return;
    }
    // {"employeeID":%1,"name":%2,"department":软件,"time":%3}
    QJsonObject obj = doc.object();
    QString employeeID = obj.value("employeeID").toString();
    QString name = obj.value("name").toString();
    QString department = obj.value("department").toString();
}
```

```
QString timestr = obj.value("time").toString();
ui->employeeIDEd->setText(employeeID);
ui->nameEd->setText(name);
ui->departmentEd->setText(department);
ui->timeEd->setText(timestr);

// 1. 直接显示/ 直接显示会显示一张正方形的图片
// QImage image(faceMat.data, faceMat.cols,
faceMat.rows,faceMat.step1(),QImage::Format_RGB888);
// QPixmap mmp = QPixmap::fromImage(image);
// ui->headLb->setPixmap(mmp);
// 2. 通过样式显示/ 可以在圆环中显示
ui->headLb->setStyleSheet("border-radius:75px;border-image:
url(./face.jpg);");
ui->widgetLb->show(); // 显示认证成功标志
}
```