

```
#include <glovehttpserver.hpp>
```

```
#include <iostream>
```

```
#include <chrono>
```

```
#include <thread>
```

```
#include <string>
```

```
#include <json.hpp>
```

```
int atoi(std::string s) {  
    try {  
        return std::stod(s);  
    } catch (std::exception &e) {  
        return 0;  
    }  
}
```

```
static std::string jsonkv(std::string k, std::string v) {  
    /* "k": "v" */  
    return ""+k+": "+v+"";  
}
```

```
class Prueba {  
    public:  
        Prueba() {  
        }  
        ~Prueba() {  
        }  
}
```

```

std::string getCollatz(unsigned num) {

    std::string out = "{ ";

    std::int i;

    //secuencia de collatz

    while (num != 1) {

        out = out + jsonkv("", std::to_string(secCollatz(num))) + ",\n"

        i++;

    }

    return out + " }";

}

size_t size() {

    return Collatzs.size();

}

/*Función que recibe un numero de tipo long int y regresa el numero de collatz*/
long int secCollatz (long int numero) {

    if ((numero%2)==1)

        return (numero=3*numero+1);

    else

        return (numero=numero/2);

```

```
    }/*termina funcion collatz*/
```

```
};
```

```
class PruebaApi {
```

```
    public:
```

```
        void get(GloveHttpRequest &request, GloveHttpResponse& response) {
```

```
            response.contentType("text/json");
```

```
            if (request.special["Num"].empty())
```

```
                response << Collatzs.getCollatzs();
```

```
            else {
```

```
                auto res = Collatzs.getCollatz(atoi(request.special["Num"]));
```

```
                if (res.empty())
```

```
                    throw GloveApiException(3, "Numero invalido");
```

```
                response << res;
```

```
            }
```

```
        }
```

```
    private:
```

```
        Prueba Collatz;
```

```
};
```

```
int main(int argc, char *argv[]) {
```

```
    PruebaApi pCollatz;
```

```

    GloveHttpServer serv(8080, "", 2048);

    namespace ph = std::placeholders;

    serv.addRest("/Collatz/$Num", 1, GloveHttpServer::jsonApiErrorCall,
std::bind(&PruebaApi::get, &pCollatz)

        );

    std::cout << "Listo"<<std::endl;

}

#include <glovehttpserver.hpp>

#include <iostream>

#include <chrono>

#include <thread>

#include <string>

#include <json.hpp>

int atoi(std::string s) {

    try {

        return std::stod(s);

    } catch (std::exception &e) {

        return 0;

    }

}

static std::string jsonkv(std::string k, std::string v) {

    /* "k": "v" */

    return ""+k+": "+v+"";

}

```

```

class Prueba {
    public:

        Prueba() {

        }

        ~Prueba() {

        }


        std::string getCollatz(unsigned num) {

            std::string out = "{ ";

            std::int i;

            //secuencia de collatz

            while (num != 1) {

                out = out + jsonkv("", std::to_string(secCollatz(num))) + ",\n"

                i++;

            }

            return out + " }";

        }


        size_t size() {

            return Collatzs.size();

        }

```

```

/*Función que recibe un numero de tipo long int y regresa el numero de collatz*/
long int secCollatz (long int numero) {

    if ((numero%2)==1)

        return (numero=3*numero+1);

    else

        return (numero=numero/2);

}/*termina funcion collatz*/

};

class PruebaApi {

    public:

        void get(GloveHttpRequest &request, GloveHttpResponse& response) {

            response.contentType("text/json");

            if (request.special["Num"].empty())

                response << Collatzs.getCollatzs();

            else {

                auto res = Collatzs.getCollatz(atoi(request.special["Num"]));

                if (res.empty())

                    throw GloveApiException(3, "Numero invalido");

                response << res;

            }

        }

}

```

```

        private:

            Prueba Collatz;

};

int main(int argc, char *argv[]) {

    PruebaApi pCollatz;

    GloveHttpServer serv(8080, "", 2048);

    namespace ph = std::placeholders;

    serv.addRest("/Collatz/$Num", 1, GloveHttpServer::jsonApiErrorCall,
std::bind(&PruebaApi::get, &pCollatz)

        );

    std::cout << "Listo"<<std::endl;

}

```