**json11**

json11 is a tiny JSON library for C++11, providing JSON parsing and serialization.

The core object provided by the library is json11::Json. A Json object represents any JSON value: null, bool, number (int or double), string (std::string), array (std::vector), or object (std::map).

Json objects act like values. They can be assigned, copied, moved, compared for equality or order, and so on. There are also helper methods Json::dump, to serialize a Json to a string, and Json::parse (static) to parse a std::string as a Json object.

It's easy to make a JSON object with C++11's new initializer syntax:

Json my\_json = Json::object {

{ "key1", "value1" },

{ "key2", false },

{ "key3", Json::array { 1, 2, 3 } },

};

std::string json\_str = my\_json.dump();

There are also implicit constructors that allow standard and user-defined types to be automatically converted to JSON. For example:

class Point {

public:

int x;

int y;

Point (int x, int y) : x(x), y(y) {}

Json to\_json() const { return Json::array { x, y }; }

};

std::vector<Point> points = { { 1, 2 }, { 10, 20 }, { 100, 200 } };

std::string points\_json = Json(points).dump();

JSON values can have their values queried and inspected:

Json json = Json::array { Json::object { { "k", "v" } } };

std::string str = json[0]["k"].string\_value();

More documentation is still to come. For now, see json11.hpp.