**API接口**

**注意：**所有AirSim API都使用NED坐标系，即+ X是北，+ Y是东，+ Z是下。所有单位均为SI系统。也就是当z轴方向速度为负值时是向上飞行。

1. **enableApiControl（bool is\_enabled）：**默认情况下，未启API控制，我们必须进行此调用以请求通过API控制。enableApiControl（true）后可以进行API控制。
2. **isApiControlEnabled（）：**如果建立API控件，则返回true。如果为false（默认值），则将忽略API调用。成功调用enableApiControl后，isApiControlEnabled应该返回true。
3. **reset（）：**将车辆重置为其原始状态。执行此命令后后，需要再次调用enableApiControl才可以进行API控制。
4. **update()：**更新控制器执行信号。
5. **armDisarm(bool arm)：**当参数arm为true时，螺旋桨旋转，准备起飞；当参数arm为false时螺旋桨停止旋转。
6. **takeoffAsync(float timeout\_sec = 20, const std::string& vehicle\_name = "")：** 起飞无人机，第一个参数是时间延迟，第二个参数是飞机的名称（当有多个无人机时才会用到）。
7. **landAsync(float timeout\_sec, const std::string& vehicle\_name)：**降落无人机。
8. **goHomeAsync(float timeout\_sec, const std::string& vehicle\_name)：**回到起始位置**。**
9. **moveByVelocityAsync(float vx, float vy, float vz, float duration, DrivetrainType drivetrain, const YawMode& yaw\_mode, const std::string& vehicle\_name)：**

以一定的速度飞行。vx,vy,vz,分别为x,y,z,方向的速度值，默认为0，当vz为负值时为向上飞行；duration为持续时间；drivetrain为动力传动参数，yaw\_mode为角速度参数，这两个一般默认即可；vehicle\_name当有多个无人机时会用到。

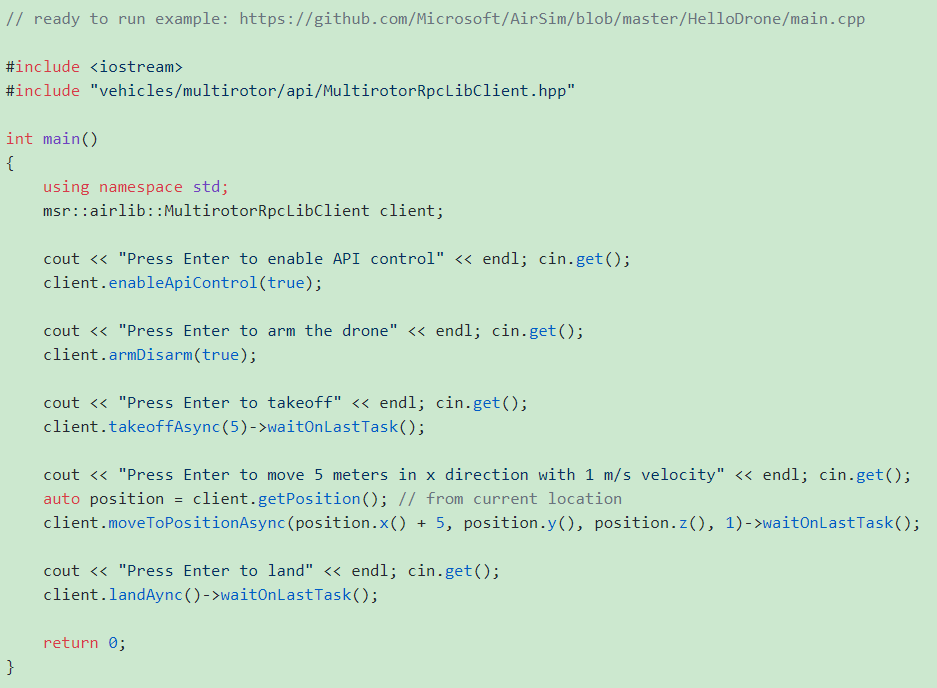
1. **moveToPositionAsync(float x, float y, float z, float velocity, float timeout\_sec, DrivetrainType drivetrain, const YawMode& yaw\_mode, float lookahead, float adaptive\_lookahead, const std::string& vehicle\_name)：**移动到目标点。x,y,z为目标点坐标。
2. ***hoverAsync(const std::string& vehicle\_name)：***进入盘旋模式。
3. **waitOnLastTask(bool\* task\_result, float timeout\_sec)：**c++接口，等当前任务执行完再在执行下一个任务。

**例：**client.takeoffAsync（） - > waitOnLastTask（）; //C++

**注：**Python接口的相同功能例子： client.takeoffAsync().join() //Python

1. **moveByAngleZAsync(float pitch, float roll, float z, float yaw, float duration, const std::string& vehicle\_name)：**
2. **moveByAngleThrottleAsync(float pitch, float roll, float throttle, float yaw\_rate, float duration, const std::string& vehicle\_name)：**
3. **moveByVelocityZAsync(float vx, float vy, float z, float duration, DrivetrainType drivetrain, const YawMode& yaw\_mode, const std::string& vehicle\_name)**
4. **moveOnPathAsync(const vector<Vector3r>& path, float velocity, float duration, DrivetrainType drivetrain, const YawMode& yaw\_mode, float lookahead, float adaptive\_lookahead, const std::string& vehicle\_name)**
5. **moveToPositionAsync(float x, float y, float z, float velocity, float timeout\_sec, DrivetrainType drivetrain, const YawMode& yaw\_mode, float lookahead, float adaptive\_lookahead, const std::string& vehicle\_name)**
6. **moveToZAsync(float z, float velocity, float timeout\_sec, const YawMode& yaw\_mode, float lookahead, float adaptive\_lookahead, const std::string& vehicle\_name)**
7. **moveByManualAsync(float vx\_max, float vy\_max, float z\_min, float duration, DrivetrainType drivetrain, const YawMode& yaw\_mode, const std::string& vehicle\_name)**
8. **rotateToYawAsync(float yaw, float timeout\_sec, float margin, const std::string& vehicle\_name)**
9. **rotateByYawRateAsync(float yaw\_rate, float duration, const std::string& vehicle\_name)**
10. **SafetyViolationType enable\_reasons, float obs\_clearance, SafetyEval::ObsAvoidanceStrategy obs\_startegy,float obs\_avoidance\_vel, const Vector3r& origin, float xy\_length, float max\_z, float min\_z, const std::string& vehicle\_name)**
11. **getMultirotorState(const std::string& vehicle\_name)**
12. **moveByRC(const RCData& rc\_data, const std::string& vehicle\_name)**

**通过调用API来控制无人机的代码示例（c++）：**



**API说明文档：**

<https://github.com/Microsoft/AirSim/blob/master/docs/apis.md>