EPSemicon
悦谱半导体

JW-Analysis SDK 说明书

(参数配置)

Ver. 1.0

目录

一、 demo 流程概览

demo 指令，分别为：

master 指令用于查询分析的总览

item 指令用于查询 item 表中信息

up 指令用于上传原始 tgz 文件

down 指令用于下载上传完成的原始文件

conn 指令用于链接服务器

post 指令用于提交分析，并存储相关信息到数据库

query 指令用于查询分析结果的存储状态

get 指令用于获取某个分析的所有分析结果

begin 指令用于开始分页查找

exit 指令用于退出 demo

help 指令用于查询以上指令解释

searchfiles 指令用于查询原始文件

delete 指令用于删除某个料的所有分析内容和原始文件

delAnalysis 指令用于删除单个分析结果

break 指令用于中断正在分析的分析进程

restart 指令用于重启分析服务器

changeRange 指令用于修改 Range 值

二、 指令详细使用说明

- master 指令

Master 指令内部现共有 success&processing, success, all, select 和 selectByMulti-conditions 这几个细分指令。其中 success&processing 用于查询成功和正在分析的分析，success 查询所有成功的分析，all 查询所有分析，select 不同条件单个值查询（JobName=xxx;LayerName=xxx;StepName=xxx;AnalysisType=xxx），selectByMulti-conditions 用于按多个条件多值查询(JobName=xxx;LayerNames=xxx,xxx;StepNames=xxx,xxx;AnalysisTypes=xxx,xxx）。

Master 使用的 API：success 指令中使用的 API 为原先的 GetFinishMasterList()，无需传入任何参数，经过处理返回 List<CheckResultMaster>类型数据，list 中主要包含所有存储成功的分析的 AnalysisGuid,JobName,StepName,AnalysisType,AnalysisStatus 等信息，success&processing 指令使用 GetFinishAndProcessingMasterList（）API，无需传入参数，all 指令使用的 API 为 GetAllStatusMasterList（），无需传入参数。Select 指令使用的 API 为 GetMasterListByPara(Dictionary<string, object> masterPara)，需要字典类型参数。selectByMulti-conditions 指令使用的 API 为 GetMasterListByCriteria(Dictionary<string, object> masterPara)，需要字典类型参数。以上 API 均返回 List<CheckResultMaster>类型数据，list 中主要包含所有存储成功的分析的 AnalysisGuid,JobName,StepName,AnalysisType,AnalysisStatus 等信息。

- item 指令

Item 指令中使用的 API 为 GetItemCountList(AanlysisTask _task)，需要传入 AanlysisTask 类型的参数，主要是标识某个分析唯一性的 uuid，经过函数处理返回 List<ItemCount>类型数据，list 中主要包含 LayerName，ItemName，1-4 不同 level 的计数。

- up 指令

Up 指令中使用的 API 为 UpdateFile(string _jobPath)，需要传入 string 类型的路径，经过函数处理返回 UpdateFileResMsg 类型数据，其中主要包含上传成功或失败的信息。

- down 指令

Down 指令中使用的 API 为 GetFilesInfo(), 无需传入参数, 经过处理, 返回 List<MFileInfo>类型数据，list 中包含 id，文件名和上传时间。

- conn 指令

Conn 指令中使用的 API 为 Connect(string _ip, int _port)，需要传入 string 类型的 IP 地址和 int 类型的端口号，连接成功返回 true 否则 false。

- post 指令

Post 指令中使用的 API 为 PostTask (MsgParams _data)，所需参数为 MsgParams 类，此类中包含分析类型，job 名，step 名，layer 名，所需参数和 range，经过函数处理，成功则返回 200，失败则返回 400，并包含成功或失败信息和 uuid。

- query 指令

Query 指令中使用的 API 为 QueryTask (AanlysisTask _task)，所需参数为包装在 AanlysisTask 类中的 uuid，经函数处理成功则返回 200，失败 400，包含成功和错误信息以及进度。

- get 指令

Get 指令中使用的 API 为 Get(AanlysisTask _task)，所需参数为包装在 AanlysisTask 类中的 uuid，经函数处理，成功则返回 200 和所有分析结果。否则返回 400 和相关信息。

- begin 指令

Begin 指令中使用的 API 为 Begin(AanlysisTask _task, List<AnalysisFilter> _filters)，所需参数为包装在 AanlysisTask 类中的 uuid 和过滤器参数。经函数处理，成功则返回 200 和指定个数的分析结果。否则返回 400 和相关信息。

- delete 指令

Delete 指令中使用的 API 为 DeleteFile(string _jobName),参数为 string 格式的料号名称（即 SDK 中输入 delete 指令后返回的料号名称中的某一个），经函数处理，成功则返回 200 以及删除成功提示，否则返回 400 和错误相关信息。

- break 指令

Break 指令中使用的 API 为 BreakTask(AanlysisTask _task)，所需参数为包装在 AanlysisTask 类中的 uuid，经函数处理，成功则返回 200 和相关信息，否则返回 400 和相关信息。

- restart 指令

Restart 指令中使用的 API 为 ReStart()，无需传递任何参数，函数运行成功返回 200，失败 400。

- searchfiles 指令

Searchfiles 指令中使用的 API 为 `GetFilesInfo` ()，无需传递任何参数，函数运行成功返回 200 和文件名列表，失败 400 和错误信息。

- delAnalysis 指令

DelAnalysis 指令中使用的 API 为 `DeleteAnalysis` (AanlysisTask _task)，所需参数为包装在 AanlysisTask 类中的 uuid，经函数处理，函数运行成功返回 200 和删除成功信息，失败 400 和相关错误信息。

- changeRange 指令

ChangeRange 指令中使用的 API 为 `UpdateRangeConfig(RangeConfig RangeModel)`，所需参数为 RangeConfig 类，经函数处理，函数运行成功返回 200 和更新成功信息，失败 400 和相关错误信息。

三、 API 接口函数详解

1. 获取所有成功的分析的 master 信息（分析情况概览）

1) **all 查询所有 master 分析情形(成功、失败、正在分析和错误等状态)使用**

EPAPIHttp 类中的 GetAllStatusMasterList()接口：

API：List<CheckResultMaster> GetAllStatusMasterList()
所需参数：无
返回内容：
<CheckResultMaster>类的 List，CheckResultMaster 类参数如下：

```
public virtual int Id { get; set; }//mastered（表 id）
public virtual string AnalysisGuid { get; set; }//分析 guid
public virtual string JobName { get; set; }//job 名
public virtual string StepName { get; set; }//step 名
public virtual string LayerNames { get; set; }//层名
public virtual string AnalysisType { get; set; }//分析类型
public virtual int AnalysisStatus { get; set; }//分析状态
public virtual string ZipGuid { get; set; }
public virtual string OrigFileName { get; set; }
public virtual string OrigFileType { get; set; }
public virtual int OverallInfoId { get; set; }
public virtual string DicVersion { get; set; }
public virtual int DelFlag { get; set; }
public virtual int CreateUserId { get; set; }
public virtual DateTime CreateTime { get; set; }
public CheckResultMaster(){CreateTime = DateTime.Now;}
```

2) success 查询所有成功的分析情形使用 EPAPIHttp 类中的 GetFinishMasterList ()接口：

API：List<CheckResultMaster> GetFinishMasterList ()
所需参数：无
返回内容：<CheckResultMaster>类的 List，CheckResultMaster 类参数同上。

3) success&processing 查询所有成功和正在分析的 master 情形使用 EPAPIHttp 类中的 GetFinishAndProcessingMasterList ()接口：

API：List<CheckResultMaster> GetFinishAndProcessingMasterList ()
所需参数：无
返回内容：<CheckResultMaster>类的 List，CheckResultMaster 类参数同上。

4) select 不同条件单个参数值查询 master 情形使用 EPAPIHttp 类中的 GetMasterListByPara(Dictionary<string, object> masterPara)接口：

API ： List<CheckResultMaster> GetMasterListByPara(Dictionary<string, object> masterPara)
所需参数：Dictionary<string, object>类型的参数，其中，string 类型的 TKey 值固定。分别为 JobName，LayerName，StepName 和 AnalysisType。格式为

JobName：xxx

LayerName：xxxx

StepName:xxx

AnalysisType=xxx

的 Dictionary 数据类型，四个参数均按需添加。参数添加方法：**四个 TKey 参数均只可添加一个 TValue。**参数说明：JobName 为模糊查询，其他均为精确查找。四个条件之间查询关键字为 and。

返回内容：<CheckResultMaster>类的 List，CheckResultMaster 类参数同上。

5) selectByMulti-conditions 多条件多值查询 master 情形使用 EPAPIHttp 类中的 GetMasterListByCriteria (Dictionary<string, object> masterPara)接口：

API ： List<CheckResultMaster> GetMasterListByPara(Dictionary<string, object> masterPara)

所需参数：Dictionary<string, object>类型的参数，其中，string 类型的 TKey 值固定。分别为 JobName，LayerName，StepName 和 AnalysisType。格式为

JobName：xxx

LayerName：xxxx，xxxx,xxxx

StepName:xxx,xxxx,xxxx

AnalysisType=xxx,xxxx,xxx

的 Dictionary 数据类型，四个参数均按需添加，各参数中，参数添加方法：**四个 TKey 参数中 JobName 只可添加一个 TValue，其他三个可添加多个 TValue 并以逗号（“，”）分隔。**参数说明：JobName 为模糊查询，其他均为精确查找。四个条件之间查询关键字为 and。

返回内容：<CheckResultMaster>类的 List，CheckResultMaster 类参数同上。

2. 获取 item 分析项参数（item 指令）

API：List<ItemCount> GetItemCountList(AanlysisTask _task)，EPAPIHttp 类中

所需参数：AanlysisTask 类型的参数，AanlysisTask 类参数如下：

```
public string uuid { get; set; }//唯一生成的 uuid
public AanlysisTask()        { }
public AanlysisTask(string _uuid){  uuid = _uuid; }
```

返回内容：

<ItemCount>类的 List，ItemCount 类参数如下：

```
public string AnalysisLayerName { get; set; }//分析层名
public string ItemName { get; set; }//分析项 key（内部排序序号）
public virtual int Lv1Count { get; set; }//Level1 分析结果计数
public virtual int Lv2Count { get; set; }//Level2 分析结果计数
public virtual int Lv3Count { get; set; }//Level3 分析结果计数
public virtual int Lv4Count { get; set; }//Level4 分析结果计数
```

3. 上传原始文件（tgz 格式）（up 指令）

API：UpdateFileResMsg UpdateFile(string _jobPath)，EPAPIHttp 类中
所需参数：原始文件路径
返回内容：
UpdateFileResMsg 格式信息。UpdateFileResMsg 类参数如下：

```
public virtual bool status { get; set; }//状态：200 成功，400 失败
public virtual string message { get; set; }//返回信息
```

4. 下载原始文件（down 指令）

API：bool DownloadFlie(string _path, MFileInfo _mFileInfo)
所需参数：路径和 MFileInfo 类型的文件信息
MFileInfo 类参数如下：

```
public string Id { get; set; }
public string Filename { get; set; }
public DateTime UploadDateTime { get; set; }
```

返回内容：成功 true，失败 false

5. 连接 Server 服务器（conn 指令）

API：bool Connect(string _ip)，EPAPIHttp 类中
所需参数：ip 地址
返回结果：成功 true，失败 false

6. 提交分析并存储分析结果（post 指令）

API：PostResMsg PostTask(AnalysisParams _data)，EPAPIHttp 类中
所需参数：AnalysisParams 类型的数据，AnalysisParams 类参数如下：

```
public AnalysisType type { get; set; } = AnalysisType.Signal;
public string job { get; set; } = "";//job 名
public string step { get; set; } = "";//step 名
public string layer { get; set; } = "";//layer 名
public dynamic param { get; set; }//动态参数，根据分析类型获取
public List<Range> range { get; set; } = new List<Range>();//range 内容
```

返回内容：
PostResMsg 类，包含参数如下：

```
public virtual int status { get; set; }//状态：200 成功，400 失败
public virtual string message { get; set; }//返回信息
public AanlysisTask task { get; set; }//uuid
```

7. 查询分析状态（query 指令）

API：QueryResMsg QueryTask (AanlysisTask _task)，EPAPIHttp 类中
所需参数：AanlysisTask 类型的参数，AanlysisTask 类参数如下：

```
public string uuid { get; set; }//唯一生成的 uuid
```

```
public AanlysisTask()        { }
public AanlysisTask(string _uuid){  uuid = _uuid; }
```

返回内容：

QueryResMsg 类，包含参数如下：

```
public virtual int status { get; set; }//状态：200 成功，400 失败
public virtual string message { get; set; }//返回信息
public ProgressMsg progress { get; set; }//进度信息
```

->进度信息 ProgressMsg 类参数如下：

```
public int value { get; set; } = 0;//进度值
public string detail { get; set; }//进度详细信息
```

8. 获取某个分析结果所有内容（get 指令）

API：GetResMsg Get(AanlysisTask _task) ，EPAPIHttp 类中

所需参数：AanlysisTask 类型的参数，AanlysisTask 类参数如下：

```
public string uuid { get; set; }//唯一生成的 uuid
public AanlysisTask()        { }
public AanlysisTask(string _uuid){  uuid = _uuid; }
```

返回内容：

GetResMsg 类，包含参数如下：

```
public virtual int status { get; set; }//状态：200 成功，400 失败
public virtual string message { get; set; }//返回信息
public AnalysisResult analysisresult { get; set; }//分析结果
```

->AnalysisResult 类内容如下：

```
public string AnalysisType { get; set; }//分析类型
public List<Ana_Layer> Layers { get; set; } = new List<Ana_Layer>();//
分析内容
```

-><Ana_Layer>类内容如下：

```
public string LayerName { get; set; }//分析层名
public List<Ana_Item> Items { get; set; } = new List<Ana_Item>();//分析
项内容
```

-><Ana_Item>类内容如下：

```
public int Analysis_CheckResultMasterId { get; set; }//master 表 id
public string AnalysisGuid { get; set; }//分析 guid
public string Analysis_LayerName { get; set; }//分析层名
public string LayerNames { get; set; }
public string ItemName { get; set; }
public int Lv1Count { get; set; }
public int Lv2Count { get; set; }
public int Lv3Count { get; set; }
public int Lv4Count { get; set; }
public  List<Ana_ItemInfo>  ItemInfos_Lv1  {  get;  set;  }  =  new
List<Ana_ItemInfo>();//Level1 具体分析结果
public  List<Ana_ItemInfo>  ItemInfos_Lv2  {  get;  set;  }  =  new
List<Ana_ItemInfo>();//Level2 具体分析结果
```

```
public List<Ana_ItemInfo> ItemInfos_Lv3 { get; set; } = new List<Ana_ItemInfo>();//Level3 具体分析结果
public List<Ana_ItemInfo> ItemInfos_Lv4 { get; set; } = new List<Ana_ItemInfo>();//Level4 具体分析结果
```

->\<Ana_ItemInfo>类内容如下：

```
public int LevelOfCheckInfo { get; set; }
public int type { get; set; }
public List<string> values { get; set; }
public List<string> symbolnames { get; set; }
public int sort_value { get; set; }
public string LayerNameForAna { get; set; }
```

备注：

分析结果结构参数及数值说明：

返回的 json 数据中，分析数据存储在 json 数组 check_info_list 项中，样例如下：[{"LevelOfCheckInfo":0,"type":1,"values":["128322915","67543571","128301636","67449976","95983"],"symbolnames":["r23.700","r23.700"],"sort_value":0}]

Symbolnames 为不同分析结果的相关 feature 信息。sort_value 内部排序值。

分析结果类型有 0 CR(叉),1 SEG(距离),2 LN(线),3 ARC(弧),4 PT(点),5 RECT(框)。

type = 0 为 CR，values 数据个数为 4 ，values 内容：1st.中心点的 X 坐标，2nd.中心点的 Y 坐标，3rd.宽度，4th.高度。

type = 1 为 SEG，values 数据个数为 5 。values 内容：1st.起点的 X 坐标，2nd.起点的 Y 坐标，3rd.终点 X 坐标，4th.终点 Y 坐标，5th.距离。

type = 2 为 LN，values 数据个数为 5 ，values 内容：1st.起点的 X 坐标，2nd.起点的 Y 坐标，3rd.终点 X 坐标，4th.终点 Y 坐标，5th.线宽。

type = 3 为 ARC，values 数据个数为 8，values 内容：1st.起点的 X 坐标，2nd.起点的 Y 坐标，3rd.终点 X 坐标，4th.终点 Y 坐标，5th.圆心 X 坐标， 6th.圆心 Y 坐标，7th.线宽，8th.弧方向 0-逆时针，1-顺时针。

type = 4 为 PT，values 数据个数为 2 ，values 内容：1st.点的 X 坐标，2nd.点的 Y 坐标。

type = 5 为 RECT，values 数据个数为 4 ，values 内容：1st.中心点的 X 坐标，2nd.中心点的 Y 坐标，3rd.宽度，4th.高度。

9. 分页查询开始函数（begin 指令）

API:FilterResMsg Begin(AanlysisTask _task, List<AnalysisFilter> _filters) ，EPAPIHttp 类中

所需参数：AanlysisTask 类的参数和 AnalysisFilter 类的列表参数。

AanlysisTask 类参数如下：

```
public string uuid { get; set; }//唯一生成的 uuid
public AanlysisTask()        { }
public AanlysisTask(string _uuid){  uuid = _uuid; }
```

AnalysisFilter 类型的参数，AnalysisFilter 类参数如下：

```
public string layer { get; set; }//层名
```

public List<FilterItem> subfilter { get; set; } = new List<FilterItem>();//子过滤条件

-><FilterItem>类型参数如下：

public string item { get; set; }//分析项名

public int level { get; set; }//0-4,4 为全部

public int count { get; set; }//单次查询多少条内容

返回结果：

FilterResMsg 类，包含参数如下：

public virtual int status { get; set; }//状态：200 成功，400 失败

public virtual string message { get; set; }//返回信息

public FilterReults result { get; set; } = new FilterReults();//结果

->FilterReults 类内容：

public List<FilterReult> result { get; set; } = new List<FilterReult>();

->FilterReult 类如下：

public string layer { get; set; }//层名

public List<RspFilterItem> filter_result { get; set; } = new List<RspFilterItem>();

->RspFilterItem 类如下：

public string item { get; set; }

public int level { get; set; }

public List<Ana_ItemInfo> check_info_list { get; set; } = new List<Ana_ItemInfo>();//ItemInfo 分析内容结果同上

10. 继续分页查询函数

API：FilterResMsg Next(AanlysisTask _task) ，EPAPIHttp 类中

所需参数：AanlysisTask 类型的参数，AanlysisTask 类参数如下：

public string uuid { get; set; }//唯一生成的 uuid

public AanlysisTask() { }

public AanlysisTask(string _uuid){ uuid = _uuid; }

返回结果：同上 FilterResMsg 类

11. 重启服务（restart 指令）

API：ReStartResMsg ReStart()，EPAPIHttp 类中

所需参数：无

返回结果：ReStartResMsg 类，成功：status：200，message：成功信息；失败：status:400,message:相关错误内容。

12. 获取 Range 默认参数

API：List<Range> GetDefaultRangsByType(string _type) ，EPAPIHttp 类中

所需参数：

string 类型分析类型，四种分别为“Drill”，“SignalLayer”，“SolderMask”，

“SilkScreen”，“BoardDrill”

返回内容：<Range>类的 List，Range 类内容如下：

```
public string name { get; set; } = "";
public string RedLv { get; set; } = "";
public string YellowLv { get; set; } = "";
public string GreenLv { get; set; } = "";
```

13. 删除料号以及所有分析（delete 指令）

API：DeleteFileResMsg DeleteFile(string _jobPath) ，EPAPIHttp 类中

所需参数：String 类型料号名

返回结果：DeleteFileResMsg 类，成功：status：200，message：delete successful；失败：status:400,message:相关错误内容。

14. 删除单个分析结果（delAnalysis 指令）

API：DeleteAnalysisResMsg DeleteAnalysis(AanlysisTask _task)，EPAPIHttp 类中

所需参数：AanlysisTask 类型参数中的 uuid，AanlysisTask 类参数如下：

```
public string uuid { get; set; }//唯一生成的 uuid
public AanlysisTask()        { }
public AanlysisTask(string _uuid){  uuid = _uuid; }
```

返回结果：DeleteAnalysisResMsg 类，成功：200，以及成功信息；失败：status:400,message:相关错误内容

15. 获取已上传文件信息(searchfiles 指令)

API：List<MFileInfo> GetFilesInfo()，EPAPIHttp 类中

所需参数：无

返回结果：<MFileInfo>类的 List，<MFileInfo>类参数如下：

```
public string Id { get; set; }//文件表 id
public string Filename { get; set; }//文件名
public DateTime UploadDateTime { get; set; }//文件上传时间
```

16. 中断分析（break 指令）

API：BreakResMsg BreakTask(AanlysisTask _task) ，EPAPIHttp 类中

所需参数：AanlysisTask 类型参数中的 uuid，AanlysisTask 类参数如下：

```
public string uuid { get; set; }//唯一生成的 uuid
public AanlysisTask()        { }
public AanlysisTask(string _uuid){  uuid = _uuid; }
```

返回结果：BreakResMsg 类，成功：status：200，message：成功信息；失败：status:400,message:相关错误内容。

17. 更新 Range 配置信息值（changeRange 指令）

API：UpdateRangeConfigResMsg UpdateRangeConfig(RangeConfig RangeModel) ，EPAPIHttp 类中

所需参数：RangeConfig 类型参数中，RangeConfig 类参数如下：

```
public int Id { get; set; }//表 id
public string RangeName { get; set; }//range 名字
public string RangeType { get; set; }//Range 隶属分析类型
public double RedLv{ get; set; }
public double YellowLv { get; set; }
public double GreenLv { get; set; }
public string RangeDescribe { get; set; }
public string RangeVersion { get; set; }
```

返回结果：UpdateRangeConfigResMsg 类，成功：status：200，message：成功信息；失败：status:400,message:相关错误内容。

网址：www.epsemicon.cpm

电话：0512-67069003

手机：19941972528（手机&微信同号）

邮箱：sales@epsemicon.com

地址：江苏省苏州市工业园区

金鸡湖大道 1355 号国际科技园三期