

智能排班接口文档

WokrTrans智能排班滴滴AIlabs算法对接一接口文档

基础数据接口

【接口一：POS数据推送接口】

接口说明：喔趣把一个周期内某客户的pos数据推送给滴滴（新增、删除、更新）

返回格式：json

请求方式：http post

请求示例：/pos/push

接口备注：此接口由滴滴提供，由喔趣一定周期（如一周）定时调用

调用频次：预计一周一次

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	时间戳
一级	token	是	String	Token
一级	cid	是	string	客户公司id
一级	data	是	array	pos具体数据
二级	did	是	String	部门id（组织id，或者门店id）
二级	gmtBill	是	String	开单时间
二级	gmtTurnover	否	String	营业结算时间
二级	money	是	String	营业额
二级	transactionNum	否	Integer	交易笔数
二级	peoples	否	Integer	人数
二级	singleSales	否	Integer	单品销量
二级	commodityCode	否	String	商品编码

二级	payment	否	String	结算方式
二级	deviceCode	否	String	收银设备号
二级	orderNo	否	String	订单号

返回参数说明：

名称	类型	说明
code	int	响应状态：可以定义几个响应状态码 比如： 100:更新成功 101:DB操作错误 102:DB更新错误 103:更新pos成功，但是更新view表失败 400:未知异常 405:非法请求 408: token校验失败，非法请求
Message	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "cid": "123456",
  "companyName": "北京奇点餐饮有限公司",
  "data": [{
    "did": "11",
    "gmtBill": "2019-05-16 12:03:22",
    "gmtTurnover": "2019-05-16 12:03:33",
    "money": "280.00",
    "transactionNum":13,
    "peoples": "3",
    "singleSales": "80",
    "commodityCode":"SPA01",
    "payment": "现金",
    "deviceCode": "A1",
    "orderNo": "002847306ac359f6016ac3cdd4c10368"
  }, {
    "did": "11",
    "gmtBill": "2019-05-17 11:03:36",
    "gmtTurnover": "2019-05-17 11:03:36",
    "money": "170.50",
    "transactionNum":5,
    "peoples": "2",
    "singleSales": "50",
    "commodityCode":"SPA01",
    "payment": "现金",
    "deviceCode": "A1",
    "orderNo": "002847306ac359f6016ac3beb3d602dc"
  }]
}
```

JSON返回示例:

```
{
  "code":100,
  "result":"操作成功"
}
```

【接口二：劳动力标准数据推送接口】

接口说明：喔趣把劳动力标准的数据推送给滴滴（新增、删除、更新）

返回格式：json

请求方式：http post

请求示例：/labor/standard/push

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	bid	是	String	业务id（全局唯一），定长32位，mysql数据库可存CHAR（32）
一级	operate_type	是	String	操作类型：新增、删除、更新（insert、delete、update）
一级	deleteBids	否	Array	批量删除bids（operate_type= ‘delete’ 时，参数只传cid和deleteBids，请以cid和bid为唯一key进行业务删除）
一级	mode	是	String	模式：单类型或者多类型（single、multi）
一级	masterType	是	String	第一业务类型（交易笔数、营业额等）
一级	slaveType	否	String	第二业务类型（交易笔数、营业额等）
一级	detailArr	是	Array	标准详情数组
二级	detailRank	是	Integer	标准详情的序号
二级	masterTypeScale	是	String	标准详情的第一业务类型刻度值（有可能是小数值）
二级	slaveTypeScale	否	String	标准详情的第二业务类型刻度值（有可能是小数值）
二级	worktimeMinute	是	Integer	标准业务单位时间：分钟数

返回参数说明：

名称	类型	说明
----	----	----

code	int	<p>响应状态：可以定义几个响应状态码</p> <p>比如：</p> <p>100:操作成功</p> <p>101:json解析错误</p> <p>102:json格式错误</p> <p>103:未包含token</p> <p>102:detail参数错误，当前mode只支持single模式</p> <p>102:detailArr为空</p> <p>102:detail格式错误</p> <p>102:detail参数错误</p> <p>102:operate_type参数错误 只允许delete、update</p> <p>102:post_json参数错误, 需要包含 bid、operate_type、mode、masterType、detailArr、cid、did</p> <p>408:DB部分操作错误</p> <p>400:其他错误</p>
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "bid": "201903292247340840341012959f0009",
  "mode": "multi",
  "masterType": "order_num",
  "slaveType": "turnover",
  "detailArr": [
    {
      "detailRank": 1,
      "masterTypeScale": "10",
      "slaveTypeScale": "100",
      "worktimeMinute": 15
    },
    {
      "detailRank": 2,
      "masterTypeScale": "10",
      "slaveTypeScale": "200",
      "worktimeMinute": 30
    },
    {
      "detailRank": 3,
      "masterTypeScale": "20",
      "slaveTypeScale": "100",
      "worktimeMinute": 30
    },
    {
      "detailRank": 4,
      "masterTypeScale": "20",
      "slaveTypeScale": "200",
      "worktimeMinute": 45
    }
  ]
}
```

JSON返回示例:

```
{
  "code": 100,
  "result": "操作成功"
}
```

【接口三：班次模板-修改接口】

接口说明：班次模板创建和修改推送数据给滴滴

返回格式：json

请求方式：http post

请求示例：/labor/shiftmod/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	bid	是	String	预测人数业务ID
一级	didArr	是	Array	部门Id
一级	opt	是	String	操作类型（update， delete）如果是delete shiftModData 可以不传
一级	shiftModData	否	Array	班次信息
二级	opt	是	String	班次模板操作类型（update， delete）
二级	shiftbid	是	String	班次BID
二级	shiftStart	是	String	班次开始时间（HH:mm）
二级	shiftEnd	是	String	班次结束时间（HH:mm）
二级	isCrossDay	是	Boolean	是否跨天

返回参数说明：

名称	类型	说明
code	int	响应状态： 可以定义几个响应状态码 比如： 100:成功 101：DB操作错误 102：JSON解析错误 103：字段校验错误 104：token校验失败 105：逻辑处理错误 400:其他未知错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6d1059c7f1c4ebb6974065f651a08c2a",
  "cid": "123456",
  "bid": "201903291241563630561001235D025d",
  "opt": "modify",

  "didArr": [
    6
  ],
  "shiftModData": [
    {
      "shiftBid": "201903291241563630561001235f0215",
      "opt": "update",
      "shiftStart": "14:00",
      "shiftEnd": "19:00",
      "isCrossDay": false
    },
    {
      "shiftBid": "201903291241563630561001234f0214",
      "shiftStart": "20:00",
      "opt": "delete",
      "shiftEnd": "03:00",
      "isCrossDay": true
    }
  ]
}
```

JSON返回示例:

```
{
  "code": 100,
  "result": "保存成功",
}
```

【接口四：组合规则-修改接口】

接口说明：组合创建和修改推送数据给滴滴

返回格式：json

请求方式：http post

请求示例：/labor/combirule/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明:

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	bid	是	String	预测人数业务ID
一级	deleteBids	否	array	批量删除以cid deleteBids为准
一级	didArr	是	Array	部门Id
一级	opt	是	String	操作类型（update， delete）如果是delete combiRuleData可以不传
一级	combiRuleData	否	Array	组合规则信息

二级	ruleData	是	String	任务bid 用逗号隔开（修改全部传递）
----	----------	---	--------	---------------------

返回参数说明：

名称	类型	说明
code	int	响应状态：可以定义几个响应状态码 比如： 100:成功 101:DB操作错误 401:json解析错误 402:json格式错误 403:未包含token 404:未包含 cid 或 didArr 或 bid 或 opt 406:存在传入项为空的情况 407:didArr不是数组 409:didArr为空 4010:未知操作类型 4011:modify操作无组合规则 4012:combiRuleData不是数组 4013:combiRuleData为空 4014:ruleData格式错误 4015:combiRuleData中有为NULL的情况 4016:rule_data key错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

```
示例
{
  "timestr":"2019-09-02 14:26:48.857",
  "token":"6d1059c7f1c4ebb6974065f651a08c2a",
  "cid":"123456",
  "bid":"201903291241563630561001235g025d",
  "didArr":[
    6
  ],
  "opt":"update",
  "combiRuleData":[
    {
      "ruleData":"201903291241563630561001235g0815,201903291241563630561001275g0815"
    },
    {
      "ruleData":"201903291241563630561001234g0414"
    }
  ]
}
```

JSON返回示例：

```
{
  "code":100,
  "result":"保存成功",
}
```

【接口五：任务详情-修改接口】

接口说明：任务详情创建和修改推送数据给滴滴

返回格式：json

请求方式：http post

请求示例：/labor/task/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	taskDetail	是	Array	任务详情信息
二级	bid	是	String	任务bid
二级	taskName	是	String	任务名称
二级	abCode	否	String	任务简码
二级	didArr	是	Array	任务部门
二级	taskType	是	String	直接工时，间接工时，固定工时（directwh，indirectwh，fixedwh）
二级	opt	是	String	操作类型（update， delete）如果是delete 二级只传bid
二级	worktimeData	是	Array	工作时间
三级	worktimeType	是	String	营业时间，自定义时间，固定时间（bisTime， customTime， fixedTime）
三级	worktimeStart	是	String	工作时间开始（HH:mm）
三级	worktimeEnd	是	String	工作时间结束（HH:mm）
二级	certs	否	Array	证书： [”健康证”， “厨师证”]
二级	taskSkill	是	Array	任务技能
三级	taskSkillBid	是	String	任务技能bid
三级	skillNum	是	String	技能值 百分比

返回参数说明：

名称	类型	说明
----	----	----

code	int	响应状态：可以定义几个响应状态码 比如： 100:成功 101: DB操作错误 102: Json解析错误 103: 字段校验错误 104: Token校验失败 105: 逻辑处理错误 400:其他未知错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6d1059c7f1c4ebb6974065f651a05a2c",
  "cid": "123456",
  "taskDetail": [
    {
      "taskName": "服务员",
      "bid": "201903291236271250271007959f0211",
      "abCode": "XS",
      "didArr": [
        6
      ],
      "taskType": "directwh",
      "opt": "update",
      "certs": [
        "健康证",
        {
          "worktimeData": [
            {
              "worktimeType": "bisTime",
              "worktimeStart": "08:00",
              "worktimeEnd": "22:00"
            }
          ],
          "taskSkill": [
            {
              "taskSkillBid": "201903291236271250271007959f0211",
              "skillNum": "80"
            }
          ]
        }
      ]
    }
  ]
}
```

JSON返回示例：

```
{
  "code": 100,
  "result": "保存成功",
}
```

【接口六：拆分规则-修改接口】

接口说明：拆分创建和修改推送数据给滴滴

返回格式：json

请求方式：http post

请求示例：/labor/shiftsplitrule/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	bid	是	String	预测人数业务ID
一级	didArr	是	Array	部门Id
一级	opt	是	String	操作类型 (update, delete) 如果是 delete shiftsplitData可以不传
一级	shiftsplitData	否	Array	组合规则信息(修改全部传递)
二级	ruleCalType	否	String	规则计算类型（目前有种：interval, shiftNum, shiftLen, worktime。班段间隔, 班段个数, 班段长度, 每天工时 单位：分钟, 班段个数：单位：个）
二级	ruleCpType	是	String	比较类型（lt, le, eq, ge, gt。小于, 小于等于, 等于, 大于等于, 大于）
二级	ruleCpNum	是	String	比较数值 分钟数
二级	dayFusion	是	Boolean	跨天融合 班次可跨天, 班段跨天计算

返回参数说明：

名称	类型	说明
code	int	响应状态：可以定义几个响应状态码 比如： 100:成功 101: DB操作错误 102: Json解析错误 103: 字段校验错误 104: Token校验失败 105: 逻辑处理错误 400:其他未知错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6d1059c7flc4ebb6974065f651a05a2c",
  "cid": "123456",
  "bid": "20190329124156363056100123kg0251",
  "didArr": [
    6
  ],
  "opt": "update",
  "shiftsplitData": [
    {
      "ruleCalType": "interval",
      "ruleCpType": "1t",
      "ruleCpNum": "120",
      "dayFusion": true
    },
    {
      "ruleCalType": "shiftNum",
      "ruleCpType": "1t",
      "ruleCpNum": "4",
      "dayFusion": true
    },
    {
      "ruleCalType": "shiftLen",
      "ruleCpType": "1t",
      "ruleCpNum": "240",
      "dayFusion": true
    },
    {
      "ruleCalType": "worktime",
      "ruleCpType": "1t",
      "ruleCpNum": "480",
      "dayFusion": true
    }
  ]
}
```

JSON返回示例:

```
{
  "code": 100,
  "result": "保存成功",
}
```

【接口七：员工可用性-修改接口】

接口说明：员工可用性修改推送数据给滴滴

返回格式：json

请求方式：http post

请求示例：/sch/availabletime/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：定时或手动

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	availabletime	是	array	可用时间数据

二级	eid	是	String	用户ID
二级	day	是	String	yyyy-MM-dd
二级	type	是	String	0:全天可排，1：全天不可排，2 部分时间可排。部分时间可排才有start， end
二级	times	否	Array	可排时间段 支持多段
三级	start	否	String	HH:mm
三级	end	否	String	HH:mm
二级	opt	是	String	操作类型（update, delete）

返回参数说明：

名称	类型	说明
code	int	响应状态：可以定义几个响应状态码 比如： 100:成功 101：DB操作错误 102：Json解析错误 103：字段校验错误 104：Token校验失败 105：逻辑处理错误 400:其他未知错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6dl059c7flc4ebb6974065f651a08c2a",
  "cid": "123456",
  "availabletime": [
    {
      "eid": "23",
      "opt": "update",
      "type": "2",
      "day": "2019-07-01",
      "times": [
        {
          "start": "08:00",
          "end": "12:00"
        },
        {
          "start": "14:00",
          "end": "02:00"
        }
      ]
    },
    {
      "eid": "23",
      "opt": "update",
      "type": "2",
      "day": "2019-07-02",
      "times": [
        {
          "start": "08:00",
          "end": "12:00"
        }
      ]
    }
  ]
}
```

JSON返回示例：

```
{
  "code": 100,
  "result": "保存成功",
}
```

【接口八： 员工技能-修改接口】

接口说明：员工技能和修改推送数据给滴滴

返回格式：json

请求方式：http post

请求示例：/sch/empskill/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填		类型	说明
一级	timestr	是	String	接口校验参数通用	
一级	token	是	String	接口校验参数通用	
一级	cid	是		String	公司id
一级	skillData	是		Array	技能数据

二级	eid	是		String	员工ID
二级	opt	是		String	操作类型（update， delete）
二级	skills	否		Array	
三级	skill	是		String	员工技能
三级	skillNum	是		String	员工技能值

返回参数说明：

名称	类型	说明
code	int	响应状态：可以定义几个响应状态码 100：删除或修改成功 101：删除或修改失败 401：json解析错误 402：json格式错误或者字段类型错误 403：未包含必填字段 405：非法请求 408：token校验失败，非法请求
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6d1059c7f1c4ebb6974065f651a08c2a",
  "cid": "123456",
  "skillData": [
    {
      "eid": "23",
      "opt": "update",
      "skills": [
        {
          "skill": "201903291241563630561001235g0815",
          "skillNum": "80"
        },
        {
          "skill": "201903291241563630561001235g0800",
          "skillNum": "60"
        }
      ]
    }
  ]
}
```

JSON返回示例：

```
{
  "code": 100,
  "result": "保存成功",
}
```

【接口九： 员工证书-修改接口】

接口说明：员工证书修改推送数据给滴滴

返回格式：json

请求方式：http post

请求示例：/sch/empcert/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	skillData	是	Array	技能数据
二级	eid	是	String	员工ID
二级	opt	是	String	操作类型（update， delete）
二级	certs	否	Array	员工证书
三级	certname	是	String	证书名字
三级	closingdate	是	String	截至日期

返回参数说明：

名称	类型	说明
code	int	响应状态： 可以定义几个响应状态码 比如： 100:成功 101:DB操作错误 102: json解析错误 103: json格式错误 104:Token校验错误 105: json参数校验错误 105:参数错误, 需要包含cid, skillData 105:参数错误, skillData需要为list 105:参数错误, skillData的每条数据需要包含eid, opt, certs 105:参数错误, opt必须在update、 delete两个值之间 105:参数错误, certs需要为list 105:参数错误, certs的每条数据需要包含certname, closingdate 400:其他未知错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6d1059c7flc4ebb6974065f651a08c2a",
  "cid": "123456",
  "skillData": [
    {
      "eid": "10",
      "opt": "update",
      "certs": [
        {
          "certname": "健康证",
          "closingdate": "2019-08-10"
        }
      ]
    },
    {
      "eid": "23",
      "opt": "update",
      "certs": [
        {
          "certname": "健康证",
          "closingdate": "2020-07-01"
        }
      ]
    }
  ]
}
```

JSON返回示例:

```
{
  "code": 100,
  "result": "成功",
}
```

【接口十：排班合规性-修改接口】

接口说明：合规性创建和修改推送数据给滴滴

返回格式：json

请求方式：http post

请求示例：/sch/compliance/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id

一级	opt	是	String	操作类型 (update, delete)
一级	ruleData	是	Array	合规规则
二级	bid	是	string	合规性规则bid
二级	ptype	是	String	规则分类 emprule(员工规则), organizerule
二级	eids	否	Array	员工ID (eids 和didArr 必须有一个)
二级	didArr	否	Array	部门Id
二级	ptypeData	是	Array	
三级	ruleType	是	String	员工合规性: timerule(工时规则), shiftrule(班次规则), daysrule(天数规则), cretrule(证书规则), 组织合规性: skilldemand(技能需求), cretdemand(证书需求), shiftdemand(班次需求)
三级	ruleCpType	是	String	比较类型 (lt, le, eq, ge, gt。小于, 小于等于, 等于, 大于等于, 大于)
三级	ruletag	是	String	员工规则: 员工班次工时: shiftTime 班次长度: shiftLen 班次间隔时数: interval 已排班次个数: schShiftNum 指定班次个数: apShiftNum 员工连续排班天数: schDayNumCon 连续休息天数: restNumCon 未排班天数: noSchNum 相同排班天数: schNumSame 休息天数: restNum 员工排班要求证书 (0: 无, 1: 有): schCert员工排班要求所需证书到期天数: certExpireDays 组织规则: 对应 cretdemand: 证书名称, skilldemand: 技能bid, shiftdemand: 班次ID
三级	ruleCpNum	是	String	比较数值
三级	cycle	否	String	统计周期day, week, month
三级	timeRange	否	String	时间范围
三级	caution	是	String	警示: (hint 提示, forbid 禁止, warning 警告)
三级	startDate	是	String	yyyy-MM-dd
三级	shiftId	否	String	ruletag = apShiftNum 必填
三级	cert	否	String	证书名称 (ruletag为schCert或certExpireDays时必填)
三级	dayFusion	是	Bool	是否跨天: true或false

返回参数说明:

名称	类型	说明
----	----	----

code	int	响应状态：可以定义几个响应状态码 比如： 100:成功 101：DB操作错误 102：Json解析错误 103：字段校验错误 104：Token校验失败 105：逻辑处理错误 400:其他未知错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6d1059c7f1c4ebb6974065f651a05a2c",
  "cid": "123456",
  "opt": "update",
  "ruleData": [
    {
      "bid": "20191216102033745033808004da0031",
      "ptype": "organizerule",
      "didArr": [
        6
      ],
      "eids": [
      ],
      "ptypeData": [
        {

```

```

        "ruleType": "skilldemand",
        "ruleCpType": "ge",
        "ruleCpNum": "8",
        "dayFusion": true,
        "ruletag": "201903291330290710291007959f0404",
        "cycle": "week",
        "shiftId": "",
        "cret": "",
        "timeRange": "1",
        "caution": "warning"
    },
    {
        "ruleType": "skilldemand",
        "ruleCpType": "ge",
        "ruleCpNum": "8",
        "dayFusion": true,
        "ruletag": "201903291236271250271007959f0231",
        "cycle": "week",
        "shiftId": "",
        "cret": "",
        "timeRange": "1",
        "startDate": "2019-07-01",
        "caution": "warning"
    }
],
{
    "ptype": "emprule",
    "didArr": [

    ],
    "eids": [
        10,
        88
    ],
    "ptypeData": [
        {
            "ruleType": "timerule",
            "ruleCpType": "gt",
            "ruleCpNum": "480",
            "ruletag": "shiftTime",
            "dayFusion": true,
            "cycle": "day",
            "shiftId": "",
            "cret": "",
            "timeRange": "1",
            "startDate": "2018-10-02",
            "caution": "hint"
        },
        {
            "ruleType": "timerule",
            "ruleCpType": "gt",
            "ruleCpNum": "600",
            "ruletag": "shiftTime",
            "dayFusion": true,
            "cycle": "day",
            "shiftId": "",
            "cret": "",
            "timeRange": "1",
            "startDate": "2018-10-02",
            "caution": "warning"
        }
    ]
}
]
}

```

JSON返回示例:

```
{
  "code":100,
  "result":"保存成功",
}
```

【接口十一：员工匹配属性-修改接口】

接口说明：员工匹配属性修改推送数据给滴滴

返回格式：json

请求方式：http post

请求示例：/sch/matchpro/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	didArr	是	Array	部门ID
一级	bid	是	String	业务ID
一级	opt	是	String	操作类型 (update, delete)
一级	deleteBids	否	Array	批量删除bids (operate_type= ‘delete’ 时，参数只传cid和 deleteBids，请以cid和bid为 唯一key进行业务批量删除)
一级	matchpros	否	Array	匹配属性
二级	weight	是	String	权重 1,2,3,4...
二级	ruleGroup	是	String	规则类型组（worktime :工时 类型， skillcert : 技能证 书， shiftrule: 班次规 则， otherrule: 其他规则）
二级	ruleDesc	否	String	规则描述
二级	ruleCpType	是	String	比较类型（lt, le, eq, ge, gt。小于，小于等于，等 于， 大于等于，大于）
二级	ruletag	是	String	规则类型（dayWt: 日排班工 时， monthWt: 月排班工 时， weekWt:周排班工 时， taskSkilled: 任务技能 熟练度， positionSkilled : 岗位技能熟练 度， positioncert: 岗位证 书要求， shiftInterval :班 次间隔， shiftNum: 班次个 数， hireAttr: 雇佣属 性， agreeOverTime: 是否同 意加班， leadNewPeople: 是 否愿意带新人）
二级	ruleCpNum	是	String	比较数值

返回参数说明：

名称	类型	说明
----	----	----

code	int	响应状态：可以定义几个响应状态码 100：删除或修改成功 101：删除或修改失败 401：json解析错误 402：json格式错误或者字段类型错误 403：未包含必填字段 405：非法请求 408：token校验失败，非法请求
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

```
示例

{
  "cid": "123456",
  "didArr": [
    6
  ],
  "opt": "update",
  "bid": "32位字符串",
  "matchpros": [
    {
      "weight": "1",
      "ruleGroup": "worktime",
      "ruleDesc": "日排班工时（小时）",
      "ruleCpType": "1t",
      "ruletag": "dayWt",
      "ruleCpNum": "240"
    }
  ]
}
```

JSON返回示例：

```
{
  "code":100,
  "result":"保存成功",
}
```

业务操作接口

【接口一：POS预测数据接口】

接口说明：喔趣通过滴滴提供的接口获取pos预测数据

返回格式：json

请求方式：http post

请求示例：/pos/result

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：

请求参数说明：

名称	必填	类型	说明
Token	是	String	服务端校验码
timestr	是	String	检验字符串
cid	是	String	公司id
predictType	是	String	按周预测或者自定义预测：“WEEK”表示按周，“CUSTOM”表示自定义预测
dayCount	否	int	一周周期长度，默认为7
Did	是	int	部门id
startDate	是	String	开始日期，用于自定义预测
endDate	是	String	结束日期，用于自定义预测
preType	是	String	预测类型：trueGMV truePeoples
smoothCoef	否	float	0-1之间的小数-平滑指数，默认0.9
cyclesCoef	否	array	加权数组

返回参数说明：

名称	类型	说明
code	int	<p>响应状态：可以定义几个响应状态码</p> <pre>{'code': 100, 'result': 'success', 'predictParam': predictParam, 'data': predictData}</pre> <pre>{'code': 101, 'result': '参数校验失败', 'predictParam': predictParam, 'data': '{'}}</pre> <pre>{'code': 400, 'result': '未知异常'}</pre> <pre>{'code': 405, 'result': '非法请求'}</pre> <pre>{'code': 406, 'result': '尚未为该部门 / 门店（didi）提供预测'}</pre> <pre>{'code': 407, 'result': '请不要预测超过 今天30天的数据'}</pre> <pre>{'code': 408, 'result': 'token校验失败， 非法请求'}</pre>
result	String	对于本次调用的说明，比如：收到数据量，异常原因等
predictParam	传的参数	输入参数
data	Dict	<pre>{ "avgDict": {...}, "smoothDict": {...}, "deep Dict": {...}, "prophetDict": {...}, "seq2seq Dict": {...} }</pre>
avgDict	Dict	<pre>{ "2019-07-12 00:00": { "value": 10, " scope": [9, 11]}</pre>

JSON参数示例：

示例

```
{
  "token": "04809897a37ad7e552dc168c835a1120",
  "timestr": "2019-08-12 20:16:07.203",
  "predictType": "WEEK",
  "dayCount": 7,
  "did": 6,
  "startDay": "2019-07-12",
  "endDay": "2019-07-19",
  "preType": "trueGMV",
  "smoothCoef": 0.9,
  "cyclesCoef": [
    0.2, 0.5
  ]
}
```

JSON返回示例:

```
{
  "code": 100,
  "result": "success",
  "predictParam": {
    "token": "04809897a37ad7e552dc168c835a1120",
    "timestr": "2019-08-12 20:16:07.203",
    "predictType": "WEEK",
    "dayCount": 7,
    "did": 6,
    "startDay": "2019-07-12",
    "endDay": "2019-07-19",
    "preType": "trueGMV",
    "smoothCoef": 0.9,
    "cyclesCoef": [0.2, 0.5]
  },
  "data": {
    "avgDict": {...},
    "smoothDict": {...},
    "deepDict": {...},
    "prophetDict": {...},
    "seq2seqDict": {...}
  }
}

{
  "2019-07-12 00:00": {
    "value": 10,
    "scope": [9, 11]
  },
  "2019-07-12 00:15": {
    "value": 10, "scope": [9, 11]
  }
}
```

【接口二：POS数据预测值修改接口】

接口说明：喔趣通过滴滴提供的接口推送页面预测pos数据的手动修改值

返回格式：json

请求方式：http post

请求示例：/pos/forecastModify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	时间戳
一级	token	是	String	时间戳
一级	cid	是	String	公司ID
一级	didArr	是	Array	部门id数组（至少传一个，传多个就需要将多部门的pos预测数据进行分摊处理）
一级	changedDataArr	是	Array	手动修改值的数组
二级	forecastDate	是	String	预测时段的日期(格式：HH:mm)
二级	startTimeStr	是	String	预测时段的开始时间（HH:mm）
二级	endTimeStr	是	String	预测时段的结束时间（HH:mm）
二级	forecastType	是	String	预测类型（营业额：trueGMV、订单量：trueOrders、客流量：truePeoples）
二级	forecastPosValue	是	String	预测的pos值
二级	changedPosValue	是	String	修改的pos值

返回参数说明：

名称	类型	说明
code	int	响应状态：可以定义几个响应状态码 比如： 100:成功 400:其他未知错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "cid": "123456",
  "didArr": [6],
  "changedDataArr": [
    {
      "forecastDate": "2019-07-12",
      "startTimeStr": "00:00",
      "endTimeStr": "00:15",
      "forecastType": "trueGMV",
      "forecastPosValue": "0.0",
      "changedPosValue": "1080.50"
    }, {
      "forecastDate": "2019-07-12",
      "startTimeStr": "00:10",
      "endTimeStr": "00:30",
      "forecastType": "trueGMV",
      "forecastPosValue": "0.0",
      "changedPosValue": "1550.50"
    }
  ]
}
```

JSON返回示例:

```
{
  "code": 100,
  "result": "操作成功",
  "data": {
    "201903292247340840341012959f0001": {
      "09:00-09:15": 30
    },
    "201903292247340840341012959f0002": {
      "09:00-09:15": 0
    }
  }
}
```

【接口三：劳动力工时预测接口】

接口说明：喔趣通过滴滴提供的接口获取劳动力工时预测数据

返回格式：json

请求方式：http post

请求示例：/labor/worktime/forecast

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	时间戳
一级	token	是	String	时间戳
一级	cid	是	String	公司ID
一级	forecastType	是	String	预测类型（任务劳动人数：0、岗位劳动人数：1、排班劳动人数：2）
一级	startDateStr	是	String	预测开始日期（格式：yyyy-MM-dd）
一级	endDateStr	是	String	预测结束日期（格式：yyyy-MM-dd）
一级	didArr	是	Array	部门id数组（至少传一个，传多个就需要将多部门的pos预测数据进行汇总处理）
一级	taskArr	否	Array	任务数组
二级	taskBid	是	String	任务bid
二级	forecastStandard	是	String	劳动力标准业务类型(也是pos数据类型)，营业额：turnover、交易笔数：order_num、客流量：peoples
二级	laborStandardBid	是	String	任务的劳动力标准的bid
二级	taskTimeArr	是	Array	任务的时间段数组
三级	startTimeStr	是	String	时段开始时间（HH:mm）
三级	endTimeStr	是	String	时段结束时间（HH:mm）

返回参数说明：

名称	类型	说明
----	----	----

code	int	<p>响应状态：可以定义几个响应状态码</p> <p>比如：</p> <p>100:预测成功</p> <p>101:json解析错误</p> <p>102:json格式错误</p> <p>103:未包含token</p> <p>104:未包含forecastType 或者 forecastType不为[node或task]</p> <p>105:forecastStandard 或者 forecastStandard不为[order_num或turnover或peoples]</p> <p>106:forecastStandard 或者 forecastStandard不为[order_num或turnover或peoples]</p> <p>107:未包含forecast_date_start 或者 forecast_date_end</p> <p>108:请保证结束时间在开始时间之后</p> <p>109:需要包含didArr，并且为list类型</p> <p>1010:需要包含nodeTimeArr，并且为list类型</p> <p>1011:需要包含taskArr，并且为list类型</p> <p>1012:需要包含nodeStandardBid、nodeBid参数</p> <p>1013:nodeBid不为空时候，nodeStandardBid必须不为空且nodeTimeArr不为空</p> <p>1014:nodeBid为空时，taskArr必须不为空</p> <p>1015:nodeBid为空时，taskBid、laborStandardBid、taskTimeArr必须在taskArr数据的每一个元素里</p>
result	String	对于本次调用的说明，比如：收到数据量，异常原因等
data	Dict	<p>工时预测的数据，</p> <p>以时段为key, 返回每个时段每个任务的预测工时</p>

JSON参数示例：

示例

```
{
  "timestr": "2019-11-27 10:31:28.707",
  "token": "07a6ef0fbf73322110993caee8788337",
  "cid": "123456",
  "forecastType": "2",
  "forecast_date_start": "2019-07-12",
  "forecast_date_end": "2019-07-18",
  "didArr": [
    6
  ],
  "taskArr": [
    {
      "taskBid": "201903292247340840341012959f0001",
      "laborStandardBid": "201903292247340840341012959f0009",
      "forecastStandard": "turnover",
      "taskTimeArr": [
        {
          "startTimeStr": "09:00",
          "endTimeStr": "09:15"
        },
        {
          "startTimeStr": "09:15",
          "endTimeStr": "09:30"
        },
        {
          "startTimeStr": "09:30",
          "endTimeStr": "09:45"
        }
      ]
    }
  ]
}
```

JSON返回示例:

```
{
  "code": 100,
  "result": "操作成功",
  "data": {
    "2019-07-12": {
      "09:00-09:15": {
        "201903292247340840341012959f0001": {
          "value": 30,
          "scope": [25, 35]
        },
        "201903292247340840341012959f0002": {
          "value": 6,
          "scope": [5, 9]
        }
      },
      "09:15-09:30": {
        "201903292247340840341012959f0001": {
          "value": 30,
          "scope": [25, 35]
        },
        "201903292247340840341012959f0002": {
          "value": 6,
          "scope": [5, 9]
        }
      }
    }
  }
}
```

```

        "value":60,
        "scope": [50, 65]
    },
    "201903292247340840341012959f0002":
    {
        "value":60,
        "scope": [55, 90]
    }
},
"09:30-09:45": {
    "201903292247340840341012959f0001":
    {
        "value":6,
        "scope": [5, 9]
    },
    "201903292247340840341012959f0002":
    {
        "value":6,
        "scope": [5, 9]
    }
},
"11:00-11:15": {
    "201903292247340840341012959f0001":
    {
        "value":6,
        "scope": [5, 9]
    },
    "201903292247340840341012959f0002":
    {
        "value":6,
        "scope": [5, 9]
    }
},
"11:15-11:30": {
    "201903292247340840341012959f0001":
    {
        "value":6,
        "scope": [5, 9]
    },
    "201903292247340840341012959f0002":
    {
        "value":6,
        "scope": [5, 9]
    }
},

```

```
    "201903292247340840341012959f0002":  
    {  
        "value":6,  
        "scope":[5,9]  
    }  
},  
"11:30-11:45":{  
    "201903292247340840341012959f0001":  
    {  
        "value":6,  
        "scope":[5,9]  
    },  
    "201903292247340840341012959f0002":  
    {  
        "value":6,  
        "scope":[5,9]  
    }  
}  
},  
"2019-07-13":{  
    "09:00-09:15":{  
        "201903292247340840341012959f0001":  
        {  
            "value":6,  
            "scope":[5,9]  
        },  
        "201903292247340840341012959f0002":  
        {  
            "value":6,  
            "scope":[5,9]  
        }  
    },  
    "09:15-09:30":{  
        "201903292247340840341012959f0001":  
        {  
            "value":6,  
            "scope":[5,9]  
        },  
        "201903292247340840341012959f0002":
```



```
{
  "value":6,
  "scope":[5,9]
},
"09:30-09:45":{
  "201903292247340840341012959f0001":
  {
    "value":6,
    "scope":[5,9]
  },
  "201903292247340840341012959f0002":
  {
    "value":6,
    "scope":[5,9]
  }
},
"11:00-11:15":{
  "201903292247340840341012959f0001":
  {
    "value":6,
    "scope":[5,9]
  },
  "201903292247340840341012959f0002":
  {
    "value":6,
    "scope":[5,9]
  }
},
"11:15-11:30":{
  "201903292247340840341012959f0001":
  {
    "value":6,
    "scope":[5,9]
  },
  "201903292247340840341012959f0002":
  {
    "value":6,
    "scope":[5,9]
```

```
        }
    },
    "11:30-11:45": {
        "201903292247340840341012959f0001":
        {
            "value": 6,
            "scope": [5, 9]
        },
        "201903292247340840341012959f0002":
        {
            "value": 6,
            "scope": [5, 9]
        }
    }
}
}
```

【接口四：劳动力工时预测值修改接口】

接口说明：喔趣通过滴滴提供的接口推送此次预测劳动力工时的手动修改值

返回格式：json

请求方式：http post

请求示例：/labor/worktime/forecastModify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	时间戳
一级	token	是	String	时间戳
一级	cid	是	String	公司ID
一级	didArr	是	Array	部门id数组（至少传一个）
一级	forecastType	是	String	预测类型（任务劳动工时：0、岗位劳动工时：1、排班劳动工时：2）
一级	forecastModifyArr	是	Array	手动修改预测工时的数组
二级	taskId	是	String	任务的bid
二级	forecastDate	是	String	预测的日期
二级	startTimeStr	是	String	时段开始时间（格式：HH:mm）
二级	endTimeStr	是	String	时段结束时间（格式：HH:mm）
二级	forecastValue	是	String	预测的工时值（分钟数，如60）
二级	editValue	是	String	手动修改的工时值（分钟数，如30）
二级	laborStandardBid	是	String	任务的劳动力标准的bid

返回参数说明：

名称	类型	说明
code	int	<p>响应状态：可以定义几个响应状态码</p> <p>比如：</p> <p>比如：</p> <p>100: 处理成功</p> <p>400: 其他错误</p> <p>101: json解析错误</p> <p>102: json格式错误</p> <p>103: 未包含token</p> <p>103:post json参数错误：cid, forecastModifyArr, forecastModifyArr [forecastDate、startTimeStr、forecastValue、editValue] 为必传参数</p> <p>103:nodeBid为必传参数</p> <p>104:nodeBid为空时， forecastModifyArr [taskBid、laborStandardBid、did、forecastStandard] 为必传参数</p> <p>105:forecastModifyArr [forecastDate、startTimeStr、endTimeStr、forecastValue、editValue] 为必传参数</p> <p>106:nodeBid不为空时候， nodeStandardBid必须存在且不为空</p>
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
paramsDict={
    "cid": "123456",
    "nodeBid": "",
    "forecastModifyArr": [
        {
            "taskBid": "201903292247340840341012959f0001",
            "forecastDate": "2019-07-02",
            "startTimeStr": "11:30",
            "forecastValue": "30",
            "editValue": "60",
            "laborStandardBid": "201903292247340840341012959f0009",
            "did": "6"
        }, {
            "taskBid": "201903292247340840341012959f0002",
            "forecastDate": "2019-07-02",
            "startTimeStr": "11:30",
            "forecastValue": "30",
            "editValue": "0",
            "laborStandardBid": "201903292247340840341012959f0010",
            "did": "14"
        }
    ]
}

paramsDict = {
    "cid": "123456",
    "nodeBid": "201903292247340840341012959f0009",
    "forecastModifyArr": [
        {
            "forecastDate": "2019-07-01",
            "startTimeStr": "09:00",
            "forecastValue": "30",
            "editValue": "60"
        }, {
            "forecastDate": "2019-07-01",
            "startTimeStr": "09:30",
            "forecastValue": "30",
            "editValue": "0"
        }
    ]
}
```

JSON返回示例：

```
{
  "code":100,
  "result":"操作成功"
}
```

【接口五：劳动力人数预测接口】

接口说明：喔趣提供参数，滴滴预测劳动力人数

返回格式：json

请求方式：http post

请求示例：/labor/manpower/forecast

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司ID
一级	forecastType	是	String	预测类型（任务劳动人数：0、岗位劳动人数：1、排班劳动人数：2）
一级	didArr	是	Array	部门id数组（至少传一个，传多个就需要将多部门的pos预测数据进行汇总处理）
一级	forecastDateStart	是	String	预测开始日期（格式：yyyy-MM-dd）
一级	forecastDateEnd	是	String	预测结束日期（格式：yyyy-MM-dd）
一级	timeInterval	是	String	间隔
一级	shiftMod	否	Array	班次模板
二级	shiftModBid	否	String	班次模板Bid
二级	didArr	否	Array	部门
二级	shiftModData	否	Array	班次模型数据
三级	shiftBid	否	String	班次对应bid
三级	shiftStart	否	String	班次开始时间（HH:mm）
三级	shiftEnd	否	String	班次结束时间（HH:mm）
三级	isCrossDay	否	Boolean	是否跨天
一级	combirule	否	Array	组合规则
二级	combiruleBid	否	String	组合规则bid
二级	didArr	否	Array	适用范围部门
二级	combiRuleData	否	Array	
三级	ruleData	否	String	组合数据
一级	shiftsplitrule	否	Array	拆分规则
二级	shiftsplitbid	是	String	拆分规则bid
二级	didArr	是	Array	适用范围部门

二级	shiftsplitData	是	Array	
三级	ruleCalType	是	String	规则计算类型（目前有三种：interval, shiftNum, shiftLen, worktime。班段间隔, 班段个数, 班段长度，每天工时 单位：分钟）
三级	ruleCpType	是	String	比较类型（lt, le, eq, ge, gt。小于, 小于等于, 等于, 大于等于, 大于）
三级	ruleCpNum	是	String	比较数值 分钟数
三级	dayFusion	是	Boolean	跨天融合 班次可跨天， 班段跨天计算

返回参数说明：

名称	类型	说明
code	int	响应状态：可以定义几个响应状态码 比如： 100:成功 101：DB操作错误 400:其他未知错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等
data	Dict	返回预测任务组和 需要的人力。以任务组合为key, 人力数为value

JSON参数示例：

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6d1059c7f1c4ebb6974065f651a08c2a",
  "cid": "123456",

  "forecastType": "0",
  "didArr": [
    6
  ],
  "forecastDateStart": "2019-07-12",
  "forecastDateEnd": "2009-07-13",
}
```

```
    "timeInterval": "30",
    "shiftMod": [
      {
        "shiftModbid": "201903291241563630561001235D025d",
        "didArr": [
          6
        ],
        "shiftModData": [
          {
            "shiftBid": "201903291241563630561001235f0215",
            "shiftStart": "14:00",
            "shiftEnd": "19:00",
            "isCrossDay": false
          },
          {
            "shiftBid": "201903291241563630561001234f0214",
            "shiftStart": "20:00",
            "shiftEnd": "03:00",
            "isCrossDay": true
          }
        ]
      }
    ],
    "combirule": [
      {
        "combiruleBid": "201903291241563630561001235g025d",
        "didArr": [
          6
        ],
        "combiRuleData": [
          {
            "ruleData": "201903291241563630561001235g0815, 201903291241563630561001275g0815"
          },
          {
            "ruleData": "201903291241563630561001234g0414"
          }
        ]
      }
    ],
    "shiftsplitrule": [
      {
        "shiftsplitbid": "20190329124156363056100123kg025l",
        "didArr": [
          6
        ],
        "shiftsplitData": [
          {
            "ruleCalType": "interval",
            "ruleCpType": "lt",
            "ruleCpNum": "120",
            "dayFusion": true
          },
          {
            "ruleCalType": "shiftNum",
            "ruleCpType": "lt",
            "ruleCpNum": "4",
            "dayFusion": true
          },
          {
            "ruleCalType": "shiftLen",
            "ruleCpType": "lt",
            "ruleCpNum": "240",
            "dayFusion": true
          },
          {
            "ruleCalType": "worktime",
            "ruleCpType": "lt",
            "ruleCpNum": "480",
            "dayFusion": true
          }
        ]
      }
    ]
  }
}
```

JSON返回示例:

```
{
  "code":100,
  "result":"操作成功",
  "data":{
    "manpowers":[
      {
        "2019-07-12":[
          {
            "taskCombination":"201903291236061290061007959f0209, 201903291241432550431007959f0213,
201903291236116910111007959f0210",
            "detail":[
              {
                "taskId":"20190605101543145043101404020054",
                "taskType":"directwh",
                "start":"10:00",
                "end":"13:00"
              },
              {
                "taskId":"201903291236116910111007959f0210",
                "taskType":"directwh",
                "start":"16:00",
                "end":"20:00"
              },
              {
                "taskId":"201903291241432550431007959f0213",
                "taskType":"indirectwh",
                "start":"14:00",
                "end":"16:00"
              }
            ],
            "scope":[3,5],
            "value": 4
          },
          {
            "taskCombination":"201903291248510440511007959f0216",
            "detail":[
              {
                "taskId":"201903291248510440511007959f0216",
                "taskType":"indirectwh",
                "start":"09:00",
                "end":"18:00"
              }
            ],
            "scope":[3,5],
            "value": 4
          }
        ]
      },
      {
        "2019-07-13":[
          {
            "taskCombination":"201903291250295220291007959f0217,
```



```
201903291241432550431007959f0213, 201903291244342970341007959f0215",
    "detail": [
        {
            "taskId": "201903291250295220291007959f0217",
            "taskType": "indirectwh",
            "start": "08:00",
            "end": "10:00"
        },
        {
            "taskId": "201903291241432550431007959f0213",
            "taskType": "indirectwh",
            "start": "10:00",
            "end": "12:00"
        },
        {
            "taskId": "201903291244342970341007959f0215",
            "taskType": "indirectwh",
            "start": "13:00",
            "end": "18:00"
        }
    ],
    "scope": [3, 5],
    "value": 4
}
]
}
]
```

【接口六：劳动人数预测-修改接口】

接口说明：喔趣将滴滴预测人数过来值，手动调整值更新

返回格式：json

请求方式：http post

请求示例：/labor/manpower/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	did	是	String	部门id
一级	forecastType	是	String	预测类型（任务劳动人数：0、岗位劳动人数：1、排班劳动人数：2）
一级	combiRuleData	是	Array	预测数据
二级	combiRule	是	String	组合规则key(bid，逗号隔开)
二级	combiRuleNewVal	是	String	修改值
二级	combiRuleOldVal	是	String	修改前的值
二级	detail	否	array	详情列表
三级	taskId	是	string	岗位Id或者任务Id或者部门Id
三级	taskType	否	string	直接工时，间接工时，固定工时（directwh，indirectwh，fixedwh）
三级	start	是	string	开始时间(hh:mm)

三级	end	是	string	结束时间(hh:mm)
----	-----	---	--------	-------------

返回参数说明：

名称	类型	说明
code	int	响应状态：可以定义几个响应状态码 比如： 100:成功 101: DB操作错误 400:其他未知错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6d1059c7f1c4ebb6974065f651a08c2a",
  "cid": "123456",
  "did": "6",

  "forecastType": "0",
  "combiRuleData": [
    {
      "2019-07-12": [
        {
          "combiRule": "20190605101543145043101404020054, 20190605101543145043101404020053",
          "combiRuleNewVal": "8",
          "combiRuleOldVal": "5"
        }
      ]
    }
  ]
}
```

JSON返回示例：

```
{
  "code": 100,
  "result": "操作成功",
}
```

【接口七：获取排班结果-接口】

接口说明：获取排班结果

返回格式：json

请求方式：http post

请求示例：/sch/cal

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用

一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	applyId	是	String	本次获取结果生成ID
一级	didArr	是	Array	部门
一级	start	是	String	开始日期 yyyy-MM-dd
一级	end	是	String	结束日期
一级	calRelyon	是	String	算法依赖（wtForecast(劳动工时预测数据)，mpForecast（第三阶段人数预测数据来源依赖））
一级	recalculation	是	String	重新计算（默认为false第一次计算新的applyId。 true为重新计算）
一级	timeInterval	是	String	时间间隔（分钟数）
一级	schType	是	String	排班类型（任务排班：0、岗位排班：1、组织排班：2）
一级	scheme	是	String	排班方案（worktime(工时最少)，emp（人员最少），effect（时效最优），violation（违反规则较少）， fillRate(满足率)）
一级	emps	是	Array	员工
二级	eid	是	String	用户Id
二级	isNew	否	String	是否为新人（yes，no）
二级	hireType	否	String	雇佣类型（full：全职， part： 兼职）
二级	agreeOverTime	否	String	是否同意加班（yes，no）
二级	leadNew	否	String	是否愿意带新人（yes，no）
二级	specialTime	否	Array	特殊时间， 时间段为不可排
三级	startTime	是	String	yyyy-MM-dd hh:mm
三级	endTime	是	String	yyyy-MM-dd hh:mm

返回参数说明：

名称	类型	说明
code	int	响应状态：可以定义几个响应状态码 比如： 100:成功 101：DB操作错误 400:其他未知错误
result	String	对于本次调用的说明，比如：收到数据量，异常原因等

JSON参数示例：

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6d1059c7flc4ebb6974065f651a05a2c",
  "cid": "123456",
  "applyId": "76542",
  "didArr": [
    6
  ],
  "start": "2019-07-12",
  "end": "2019-07-18",
  "schType": "task",

  "calReylon": "wtForecast",

  "timeInterval": "30",
  "scheme": "worktime",
  "emps": [
    {
      "eid": "23",

      "isNew": "yes",

      "hireType": "full",

      "agreeOverTime": "yes",

      "leadNew": "yes",
      "specialTime": [
        {
          "startTime": "2019-07-12 14:00",
          "endTime": "2019-07-12 18:00"
        },
        {
          "startTime": "2019-07-15 09:00",
          "endTime": "2019-07-16 18:00"
        }
      ]
    }
  ]
}
```

JSON返回示例:

```

{
  "code":100,
  "result":"操作成功",
  "data":{
    "schemes":[
      {
        "worktime":""
      }
    ],
    "applyId":"76542",
    "status":"success",
    "cal":[
      {
        "eid":"23",
        "schDay":"2019-07-12",
        "allWorkTime":480,
        "workUnit":[
          {
            "type":"task",
            "outId":"任务ID",

            "shiftId":"班次模板ID",

            "startTime":"2019-07-12 09:00",
            "endTime":"2019-07-12 12:00",
            "workTime":180
          },
          {
            "type":"task",
            "outId":"任务ID",

            "shiftId":"班次模板ID",
            "startTime":"2019-07-12 13:00",
            "endTime":"2019-07-12 18:00",
            "workTime":300
          }
        ]
      },
      {
        "eid":"23",
        "schDay":"2019-07-13",
        "allWorkTime":480,
        "workUnit":[
          {
            "type":"task",
            "outId":"任务ID",

            "shiftId":"班次模板ID",
            "startTime":"2019-07-13 09:00",
            "endTime":"2019-07-13 12:00",
            "workTime":180
          },
          {
            "type":"task",
            "outId":"任务ID",

            "shiftId":"班次模板ID",
            "startTime":"2019-07-13 13:00",
            "endTime":"2019-07-13 18:00",
            "workTime":300
          }
        ]
      }
    ]
  }
}

```

如果返回有异常

```

{
  "code":102,
  "result":"规则有冲突" //先模糊指定下异常信息
}

```

【接口八：排班结果修改推送】

接口说明：排班结果修改推送

返回格式：json

请求方式：http post

请求示例：/sch/cal/modify

接口备注：此接口由滴滴提供，由喔趣调用

调用频次：即时调用

请求参数说明：

层级	名称	必填	类型	说明
一级	timestr	是	String	接口校验参数通用
一级	token	是	String	接口校验参数通用
一级	cid	是	String	公司id
一级	editData	是	Array	如有修改本次结果全部推送包含未修改
二级	eid	是	String	用户ID
二级	schDay	是	String	排班天yyyy-MM-dd
二级	allWorkTime	否	String	当天排班工时 分钟数
二级	workUnit	是	Array	一天的工作单元
三级	type	是	String	task: 任务
三级	outId	是	String	工作单元对应ID 类型为task 此ID 为 任务ID
三级	startTime	是	String	yyyy-MM-dd hh:mm
三级	endTime	是	String	yyyy-MM-dd hh:mm
三级	workTime	否	String	工作单元 分钟数

返回参数说明：

名称	类型	说明
code	int	响应状态：可以定义几个响应状态码 比如： 100:成功 101:DB操作错误 102: json解析错误 103: json格式错误 104:Token校验错误 105: json参数校验错误 105:参数错误，需要包含cid, applyId 105:参数错误, editData需要为list 105:参数错误, editData的每条数据需要包含eid, schDay, allWorkTime, workUnit 105:参数错误, workUnit需要为list 105:参数错误, workUnit的每条数据需要包含type, outId, startTime, endTime, workTime 400:其他未知错误

result	String	对于本次调用的说明，比如：收到数据量，异常原因等
--------	--------	--------------------------

JSON参数示例:

示例

```
{
  "timestr": "2019-09-02 14:26:48.857",
  "token": "6d1059c7f1c4ebb6974065f651a05a2c",
  "cid": "123456",
  "editData": [
    {
      "eid": "23",
      "schDay": "2019-07-12",
      "allWorkTime": "480",
      "workUnit": [
        {
          "type": "task",
          "outId": "任务ID",
          "startTime": "2019-07-12 09:00",
          "endTime": "2019-07-12 12:00",
          "workTime": "180"
        },
        {
          "type": "task",
          "outId": "任务ID",
          "startTime": "2019-07-12 13:00",
          "endTime": "2019-07-12 18:00",
          "workTime": "300"
        }
      ]
    },
    {
      "eid": "23",
      "schDay": "2019-07-13",
      "allWorkTime": "480",
      "workUnit": [
        {
          "type": "task",
          "outId": "任务ID",
          "startTime": "2019-07-13 09:00",
          "endTime": "2019-07-13 12:00",
          "workTime": "180"
        },
        {
          "type": "task",
          "outId": "任务ID",
          "startTime": "2019-07-13 13:00",
          "endTime": "2019-07-13 18:00",
          "workTime": "300"
        }
      ]
    }
  ]
}
```

JSON返回示例:

```
{
  "code": 100,
  "result": "成功",
}
```