v5.20Bybit Pay Open API

Version	Date	Changes
1.0	2025-02-10	• Initial
5.0	2025-02-17	• Modify the state machine
5.0	2025-02-17	• Add "Pay Id" to the Payment Notify demo • Add "Authentication" description to Payment Notify • Add "MccCodes" explanation of goods details
5.0	2025-03-03	• Change `clientId` to not required when initial a refund or query the order status • Move filed `goods` into `PayOrder` structure • Move filed `customer` from `PayOrder` structure • Rename `orderAmount` to amount for all `PayOrder` type • Change request body type from json array to json object when creating a refund order • Add `merchantRefundNo` when creating a refund order • Add error code `50000` <Bybit internal error> • Add error code `400004` <Order not found> • Add error code `500105` <Order not paid> • Change `regionTime` to `registerTime` in <Customer> type • Add Type <RefundOrder>, remove fields of refund from <PayOrder> • Remove all <Customer> and <Good> data from response structure • Add <Signature Algorithm> description • Change notify request body • Change timestamp from millisecon precision to second precision
5.0	2025-03-31	• Add 5.1.4 Payment mock • Add 5.3 Payout
5.0	2025-04-16	• Add 5.4 Settlement

5.0	2025-05-09	• Add payout status
5.0	2025-05-13	• Refactor pay status mock path from `/v5/bybitpay/pay-status/mock` to `/v5/bybitpay/paystatus/mock`
5.0	2025-05-21	• Add webhook support merchant public key
5.0	2025-05-27	• Refactor payment type from `BYBIT_PAY_MERCHANT_PAYOUT` to `MERCHANT_PAYOUT`

5.0	2025-05-28	• **Add customer uid in API** `/v5/bybitpay/pay_result` **、Payment Notify**
		"**customer**":{
		"uid":"123456",
		"externalUserId": "test@gmail.com",
		"userName": "john li",
		"registerTime": "1739178959",
		"kycCountry": "AUS"
		}

5.0	2025-06-01	Add crypto scale
5.0	2025-06-30	Update signature algorithm
5.01	2025-07-04	Add 5.1.5 paymentConsult api And support fiat or crypto create order ,settlement diff currency
5.01	2025-07-04	Add payout mcc Required
5.01	2025-09-03	Add error codes

5.01	2025-09-09	Add `remarks` in API `/v5/bybitpay/create_pay` and `/v5/bybitpay/pay_result`
		"customer":{
		"externalUserId": "test@gmail.com",
		"userName": "john li",
		"registerTime": "1739178959",
		"kycCountry": "AUS",
		"remarks":""
		}

💡 Join Bybit Pay's ecosystem for low-fee crypto and fiat transactions. Reach 60M+ users and revolutionize global payments with seamless Web2 & Web3 integration.

Learn More | Partner with Us

1. Authentication

- Reference documentation: Bybit API Guide
- Create a new key: Bybit Secret Key

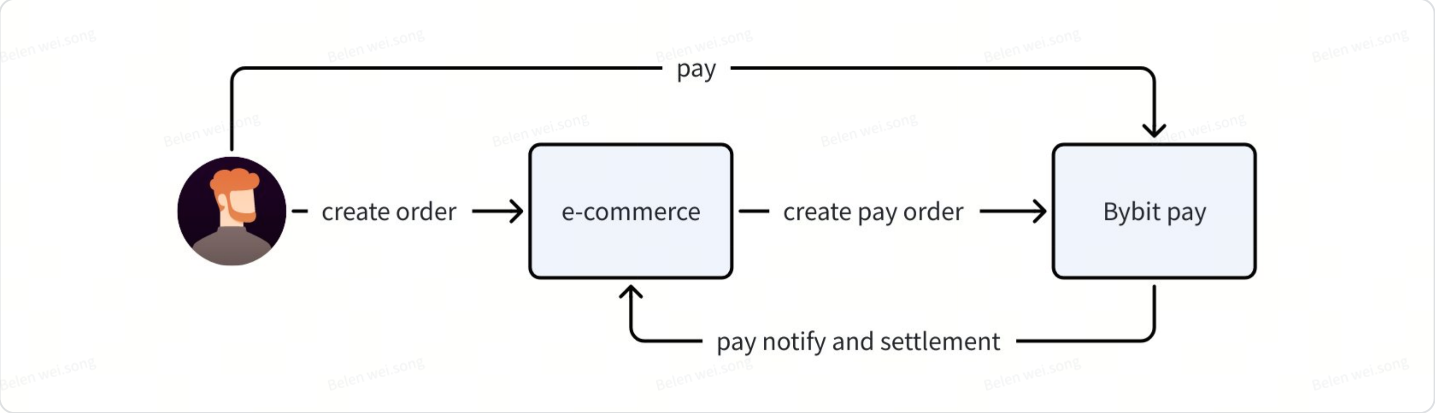
2. Version

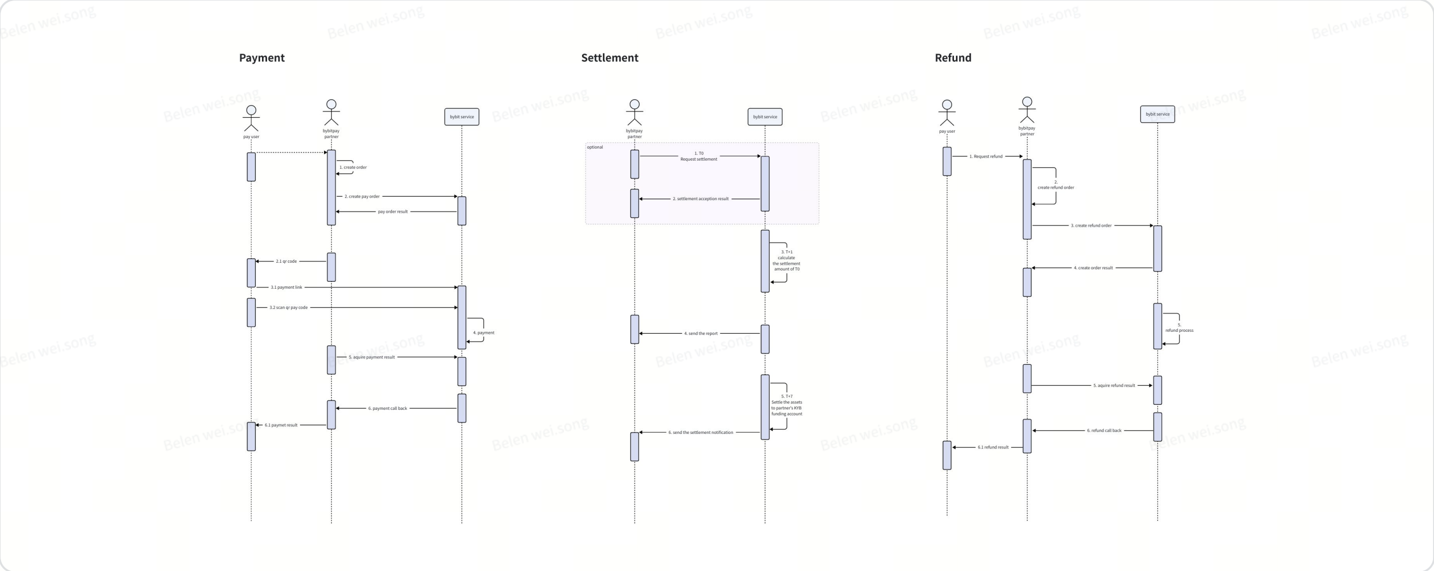
http head add version， default v5.00

代码块

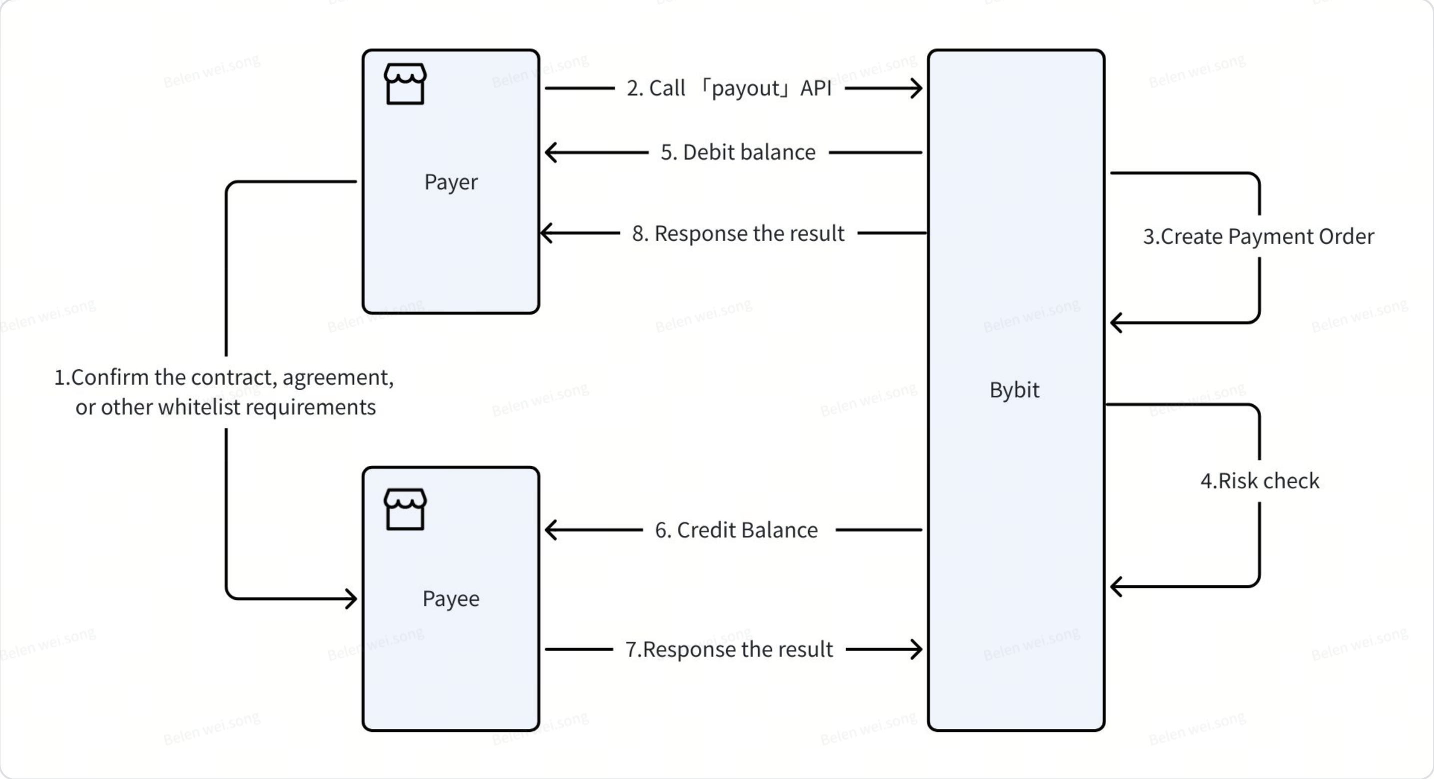
```
curl --request POST \
  --url https://api.bybit.com/payout \
  --header 'Version: 5.00' \
  --header 'Content-Type: application/json' \
  --data '{
}'
```

3. Flowchart

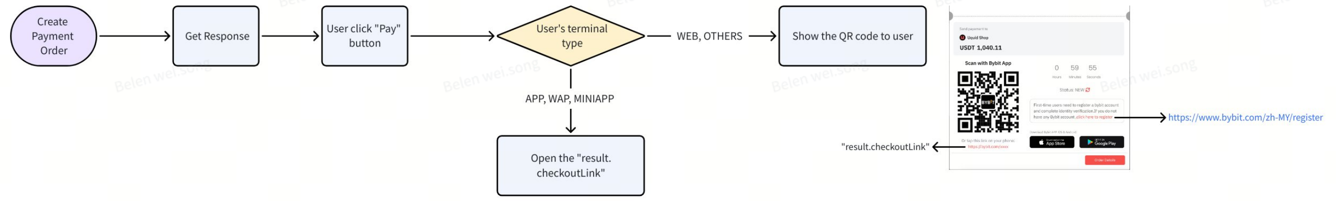




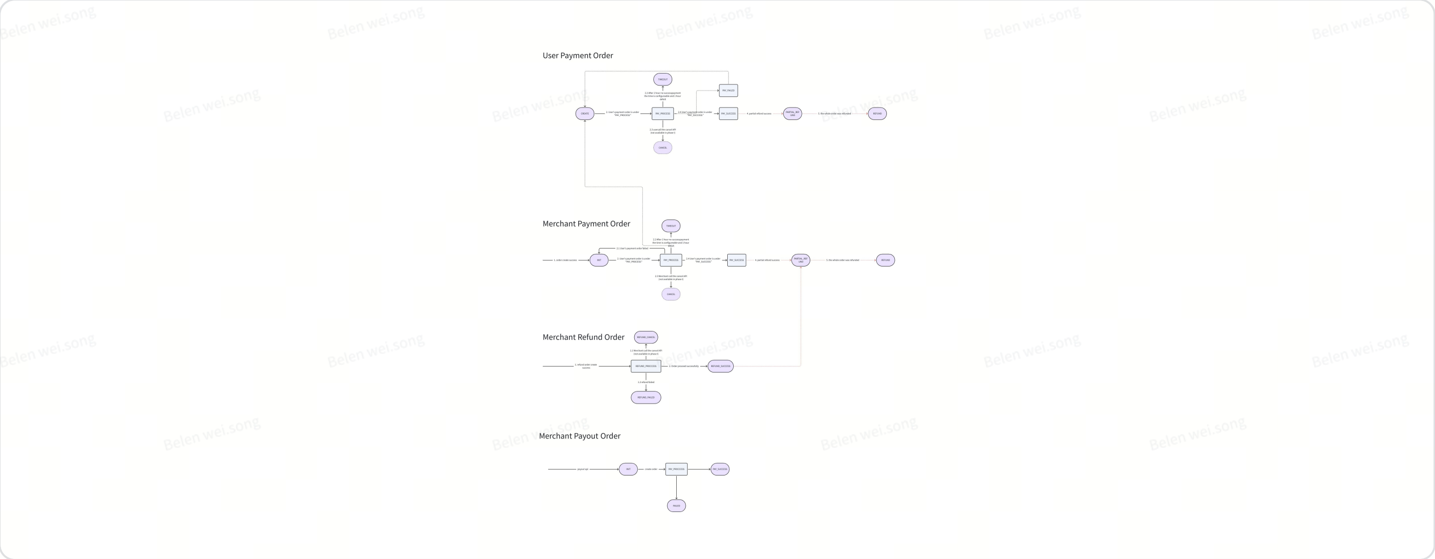
payout Flowchart



4. Users' Guide after creating payment order



5. Order Status


User Payment Order
TIMEOUT
PAY_FAILED
CREATE
2. User's payment order is under "PAY_PROCESS"
2.2 After 1 hour no successpayment the time is configureable and 1 hour defult
PAY_PROCESS
2.4 User's payment order is under "PAY_SUCCESS"
PAY_SUCCESS
4. partial refund success
PARTIAL_REFUND
5. the whole order was refunded
REFUND
2.3 usercall the cancel API (not available in phase I)
CANCEL
Merchant Payment Order
TIMEOUT
2.2 After 1 hour no successpayment the time is configureable and 1 hour defult
2.1 User's payment order failed
1. order create success
INIT
2. User's payment order is under "PAY_PROCESS"
PAY_PROCESS
2.4 User's payment order is under "PAY_SUCCESS"
PAY_SUCCESS
4. partial refund success
PARTIAL_REFUND
5. the whole order was refunded
REFUND
2.3 Merchant call the cancel API (not available in phase I)
CANCEL
Merchant Refund Order
REFUND_CANCEL
1.1 Merchant call the cancel API (not available in phase I)
1. refund order create success
REFUND_PROCCESS
2. Order proceed successfully
REFUND_SUCCESS
1.2 refund failed
REFUND_FAILED
Merchant Payout Order
payout api
INIT
create order
PAY_PROCCESS
PAY_SUCCESS
FAILED

6. API Documentation

6.1 Scan Payment

6.1.1 Payment Creation

- **Data Type:** JSON (content-type: application/json)
- **Request Method:** POST
- **Path:** `/v5/bybitpay/create_pay`

Request Body:

Field Name	Type	Required	Description
merchantId	string	Yes	Onboarding merchant UID
merchantName	string	No	If not filled in, we will show users the name of your company doing KYB on the Bybit platform
clientId	string	No	Merchant Client ID
paymentType	string	Yes	Payment Type
customer	<Customer Type>	No	Customer info
merchantTradeNo	string	Yes	Merchant Order Number
quotationId	string	No	Return Result quotationId to create order
goods	array< `Good Type` >	Yes	Product Description
orderAmount	string	Yes	Order Amount

currency	string	Yes	Order Currency (uppercase, e.g., EUR, USD, USDT)
currencyType	string	Yes	Currency Type (`fiat` or `crypto`)
successUrl	string	Yes	URL to return after successful payment (max 256 characters)
failedUrl	string	Yes	URL to return after failed payment (max 256 characters)
webhookUrl	string	Yes	API URL to call when order is successful or failed (max 256 characters)
orderExpireTime	int64	No	Order expiration time in UTC timestamp (second precision), default is 1 hour, max is 1 hour
env	`Env Type`	Yes	Transaction Source
riskInfo	<RiskInfo>	No	Risk info for order pay
remark	string	No	Payment remark ,max 50 characters

Request demo

代码块

```
{
    "currency": "USDT",
    "currencyType": "crypto",
    "env": {
        "browserVersion": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36",
        "device": "xxde",
        "ip": "127.0.0.1",
        "terminalType": "APP"
    },
    "failedUrl": "",
    "customer":{
        "externalUserId": "test@gmail.com",
        "userName": "john li",
        "registerTime": "1739178959",
        "kycCountry": "AUS"
    },
    "goods": [
        {
            "shoppingName": "test good",
            "mccCode": "1520",
            "goodsName": "test",
            "goodsDetail": "test"
        }
    ],
    "merchantId": "305142568",
    "merchantName": "uquid",
    "merchantTradeNo": "841e4ba2-c7ec-40b1-a071-a2a45de7bd00",
    "orderAmount": "100",
    "orderExpireTime": 0,
    "paymentType": "E_COMMERCE",
    "successUrl": "www.bybit.com/successUrl",
    "webhookUrl": "www.bybit.com/successUrl"
}
```

Response:

Field Name	Type	Required	Description
retCode	number	Yes	Return Code, 100000: success, not 0: failed, See<Error Code>
retMsg	string	Yes	Return Message
result	object	Yes	Return Result information
result.payId	string	Yes	Pay-order ID created successfully
result.terminalType	string	Yes	Transaction source (APP, WEB, WAP, MINIAPP, OTHERS)
result.expireTime	number	Yes	Order expiration time in UTC timestamp (second precision)
result.checkoutLink	string	Yes	Unified checkout link in Bybit, redirect for payment
result.qrContent	string	Yes	QR code content (base64 Image, valid for 1 hour), developers generate QR image
result.order	`PayOrder`	Yes	Bybit PayOrder Detail
result.customer	<Customer Type>	No	Customer info

Response demo

代码块

```

{
    "retCode": 100000,
    "retMsg": "success",
    "result": {
        "payId": "01JXHW8QAEZXSA5HTJYT3AS357",
        "terminalType": "WEB",
        "expireTime": 1749729103,
        "checkoutLink": "bybitapp://open/route?targetUrl=by-
mini%3A%2F%2Fpay%2FscanResult%3ForderNo%3D01JXHW8QAEZXSA5HTJYT3AS357",
        "qrContent":
"data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAZAAAAGQCAIAAAAP3aGbAACAAElEQVR4
nOz9B3gV1fb4D2fOnF7TKyQkNBMp0jvSVXpTWgS8l35RVCAUAfFSLnBBQaQXpStCpEZBRAIxgBQpgVA
SEkJCej0pp82c9/lm3n9+c9c+ZDaTQzm4P8997mMWe/Zes2fPOufstddaUrvd7kYgEAAiugORFK0AgEA
i4EINFIBBcBmKwCASCy0AMFoFAcBmIwSIQCC4DMVgEAsFlIAaLQCC4DMRgEQgEl4EYLAKB4DIQg0UgE
FwGYrAIBILLIMVpxLKs0Wi02WzPXh8BJBKJXq+naZovNJlMZWVlfAlUXq9XirFujscSktLzWaziAsx
FXY6Wq1WoVAINisvL6+oqBDRP0VROp1OJpNV38xut5eUlGDRL5QKpXqdq9H+LNEZpmlar9eDZqI
Vxken08nlchEXMgxTULCsmz1zSQSiU6nE1ycdrvdaDRarVYRmjicYRRMhZ8DmArjGqySkpKJEyfevX
vXGbrVCF9f3+3+++++4wMJAvjI6OXrFiBV+i1+s3127N1945
MiRpUuXOks3h3zxxRcDBgwQbPbdd99t3rxZRP9arXbTpk1hYWHVNzOZTKNHj75//76IIfr37//Fxc
0/GFX375ZXR0NF8SHBy8SHBy8fft2b29vvnDXrl0bNmwQoTAmNE2vXh7/xdmDlbPzkOTe9DOIxQ4P9Q
Pf5/5yx/jOz/4OeDbvxQjEmSE-this_is_unread_safe_guard
```

```
AwIGEY5vLlyxcuXOALVSpV165dcb5Q4KBWq1G/Pkp2djZ60g2FYZgrV67Ex8fzhXK5HOeLg8Fg6NKlC
xAeOHAAZ+ratGkj2IaLQs/KyhJsVlBQcObMGZwOnYjVar148aJos+jcnw5kD4tAILgMxGARCASXgRgs
AoHgMhCDRSAQXAZnbrq/KIKCgsCeqFKpvHv3LkgdmZWV9eabb/IlLMvevHkThBwbDIYmTZoAL6HooDa
5XN6sWTOVSsUXZmRk3L9/X1yHoikrK7t27RpwuqWmpoJmMpmsWbNmarWaL9TpdL///jvorbi4+FnqWy
MyMzOBwmazOSwsDBzOCAoKOnv2LHjWKHq9vkmTJsCZ0KBBA3QnHiU0NBRIZDJZq1atfH19Me5DGLvdf
uvWrby8PKf09vLzKhis/v37v/3223xJUVHR0KFDwZGFd95558iRI3yXsNlsfvfdd4Gv5/XXXz9w4ABw
PIn2y3p6em7duhWs2q1bt3788cfiOhRNenr6qFGjgFsTPTTg4eGxZcsWkIdg165dwOnOnc15lvrWiFO
nToFE8hqN5ocffgAZEX777bchQ4YInkRr0aLFkSNHQMbxKVOmTJgwQVAT1Gfq7u6+YcMGZ/nOWJYdOX
LksWPHnNNLby8+rYLBklfAlZrPZYrGAQ1IMw2g0Gr7BoisBvdE0rVarcTzlOFAUpVKpQG/iirLUEC47h
eDBMYcK0zTtrBNnzweGYYDC3AlGcF8ymayiokLwPI3DFOyKSkToxs2wiAsdwrKsE6tDvfyQPSwCgeAy
EINFIBBcBmKwCASCy0AMFoFAcBleze06jUazdOlSUEw0MzNz1KhRfIlEIunTp88HH3zAF+bn50+aNAl
Euk6ZMgUnrg2lsLDw448/1mq1fGFISMjevXtF9CaRSNBc1Xfv3l22bJlgbU6NRrN8+XLBfWKz2bx8+X
JQST85ORlHvatXr0ZGRopLzzBu3DgQrZmZmfnFF1+AwxODBw9+9913+RKDwDbwbq1SpwhCU2NnbTpk2Cg
zZt2nTXrl2CDjsvLy9wzsMh2dnZCxcuFMzprtfr582bB4qBo5hMpXLl9+5c4cvbNy48YwZM16zI0+Zl
ASePssOc7kOHDgXZr8vLy7t16yZOjcaNG+PuJudtKkSeJuwSFxcXHAIDqkYcOGWVlZgr1lZmbWq1dP3H3VhM2bNwNN7t27hx5/
9/Ps6cbN++HVyo1Wrj4uLEzjEWSUlJOPVBfHx8EhISBHtzmNO9Z8+eXHqZKhiGGThwIGhWk5zuM2fOB
M1yc3MbN26M/ST/h27dupWXl/N7Y1kWrXuCn9Od/CQkEAguAzFYBALBZSAGi0AguAzEYBEIBJfBmV5C
iqJCQ0PF7c/Vr19fMAb1SeTl5eFUvn348CFOekm1Wv3666+DBNtGoxFEJiYnJ+NkQ5dKpXr1gVuHXS
n0yHFxcVpaWmCOicnJ6NZvWvVquXh4cGXBAQE3Lt3D5S59qsEKNygQQMQO1JQUABK9UokktDQUBzfGU
p5eXlKSgrwzaWnp4MZzsrKqlevHhokDJpJdKwsDCc1O8oRqMxNTVVcIbVanoaKjg+pTL5a+99pq3t
3f1zTw8PESHphqNxoSEBP5ystvt7u7u4KWrV6+e6Lchabp+vXri7s2NDTUuYUanGmwFArFqlWrxGU6
pmladEL36OjouXPnCjYTzA7O0bRp05iYGLCIo6KiFixYwJcwcDIOTrsDLy2v37t116tThCzHfrvj4+LF
jxwqWWEDrF3C+npEjR/48eP/IlDx48eP/99/Pz8/nCWbNmAVesp6fn7ffn7t27waC7IHALNE2LLLhvDPVZBg1VcXIzz3ZymaU9PT3
GaYIKuQ66ij7gHgYnDJfGiIHtYBALBZSAGi0AguAzEYBEIBJeBGCwCgeAyiN90v3HjRlRlRU1DMtLi+6Z
qRKpRo/fnxQUBBfYvS//nxQUBBfeOvWrZ07d2i1WryGpgm5FPIPvn
kE7DbXVZWBppptdopU6aI28S9ffv2zp07+TcrkUhGjhx56nz5MmTgd/p9OnTv
n2u+++AzOsUChAM7vdfvz48cOHD/OFMpns3XffRfQkdHRoI6sp6fn5MmTgd/p9OnTv
/zyC1+i0WgmzUAQ7z11lsgM/X9+/e/fZbQSdenr6li1bgEelf//+HTt2FLyv
M2fOgCXhXOx201pu/emuF8Rh8PMLwWHw85o1a0AzDw+Py5fh5yZII1FlGRgZaNhkNfi4sLES
PCPTv399qtfKbmc1m4NV2c3Nr3759SUkJ6HDS2LGgmcPg5/nz54+9+vj4eN
Ds6NGjIF2vVCo9evevQoaBYfH6/X60X0vvbbb5vNZzq9Xav
39/0KxVq1aFhYWgw0mTJoFmM4fBz+hHqUajiiY2NBc1OnDgBgBPpkoioqOjgbNLl++
jPr11qxZA5o5DH5+IZDgZwKB8ApCDBaBQHAZiMEiEAguAzFYBALBZcDyElIU5ePj+/7I9kPj4l2T
hcrk8ICAAVLUzmUMvg+regoMDX1xfUQ2RZFjTLzs7GiZFjTEz9vbG8dvarDD
bMzEygDMgszPnmsrOzQVQKI79JBQKRWGpi4/P9I
lAul4PeysvL4L9Xo9mDoTp46BotYePFnlPw4ev5dgOaoP2PqVIGlKbQNCKA2bLiO
NC26pLlz8fHxwTxvgGWwDAbDh7zxxvXCdDAbDjh07MIOHnyk0TeNkoa0Jb7zxxqlTp4A3fUklElISMiuXbuAL2bLi0
dOnTgSxiGycrKEhxUp9Nt3F3boV1FKOi4vr0qULf+qVJSf5asXLmS9S30yhUhS+vXrx94
UFBcOGDQOGJiDJ9GhQ4ezZ8/+/yp44zxDj/jzzz/G++fyjzzzz+DB
6FSqf773/82atQICMGd0jS9/99Pnyznzyzyzyzyzyzyz
Mnfu3MWLF/MlFosFGcuiC3F55EFBAQcPnYdCSmE5HL5ZhhyFgGi6bpl+Hr1fNBqVSGhIQAocGYxyfAGGAg==
CgoJA8hO73Q6aYSKRSFBDfP369bS0NP56kslker0e5H7AxGazgXNeDmFZFidXj0PUajU6dZgUVMKXyG
SywMBAcF4JhaZpcOCOe4jgQWi1Wg8PD8GpoygKTWiTk5OTkZGBk5wDB4czbDKZxD1WNze33NxccRdKp
VLBWhgvG2QPi0AguAzEYBEIBJeBGCwCgeAyEINFIBBcBmdmHLXZbEePHn306JGIa729vQcNGgRSicfF
xV29epUv0Wg07777LhqJ5kQyMjKOHDkCvE61a9f+8MMP+RKZTLZnzx7gdLt8+bK4Qc1m86FDh0DC9bK
ysilTpoCW165du3//Pl8SEBAwYMAAQRe4u7v7kCFDxGVhT0hI+P333/kSi8Xy448/Xrp0SURvqampTv
Q4l5eXHzx4EBy8KCkpAc+Loqi4uLgrV65U35tEIunduzcI//Tz85swYYJgbW0vLy80CrVly5ZAE4e0b
NkSSCwWy5EjR8D2/J07d1CnXu/evevWrcuX5OTkHDp0CDidY2NjBdVwiFKpHDx4sGA8vNVqPXr0KI5j
B8Xh6+8YnIBDTF5U5eeXJ/gZk5cn+BkTNPj5OVCvXr3MzExB3WoS/IxC0/ShQ4fEzZJzcVj5GQU/+Fk
0Xl5e169fF1S4tLS0S5cu4oYgwc8EAuEVhBgsAoHgMhCDRSAQXAZisAgEgsuA5SVkWba0tFQwiNdsNi
uVSnG7fWq1uri4GCTwpSgK9KbT6YxGI4jhAIHKnMIlJSWgWWlpKYgQ5LY2QbOKigqDwQCcblarFe1Nr
9fjBDajiHPVcXAhJnyJSqUqKioCbk2pVAqaaTSa4uJiEMKGicNAX51OB/KLOhe9Xo8GxFZUwpcUFxdr
NBpwsw5nWKfTCS5OiUSCOUVlZWXADVcTFAoF6upBn7XVagUFaLndbrA4S0pKcIKfJRKJTqcDL53JZAI
z7PBtksvlaLVKFKlUqtVqBQObdTodZvAzhVMhsqioaPLkySCVOIpMJps6dWp4eDjOwID09PRVq1aBF2
PIkCFvvfUWX1JYWLhixQoQd5qTkwOcqVxxbbBqi4qKHjx4wJdQFFW3bl1wSKJ+/fofffQRyDGwbdu2i
xcv8iUBAQHTp08Xd8DCx8cnODgY6NarVy9wSqB///4HDx7kGwWWZZOTk0GKhaSkpDVr1oCXZ8yYMSD6
Nysr67///S9mGDMgLy8vLS2NL9FoNGvXrm3SpImI3jBRKBQNGzYE5mPz5s2bNm3iS3Q63cyZM0HIoZe
XFwjNYxjm/v37gikWKIoKCwtDQ6xRli5devDgQbxbEaZ3796LFi3iSxiGSUpKAubpwoULn3zyCfhwQh
UuLy+/f/++YFSzv7//9u3bQezkjh071q5dy5fQNN2gQQNw5uCtt94CBQfKysr69u0LZLbt28DSzRs2DDQW0ZGxr179wQD
jBmGuXPnjuCgdrs9KSkJCDUaTZMmcTcnx+rVq8GJsIiINdffx0NlH2mSCQStCa7yWS6efMmWNlTp04
FU3fnzp3Ex8SCnByBynaELT9GuvvSbuWdeErKws8CB8fX3r1q372muwPmdmjB7HQpCagdTpomk
aPa+Tn56NFs8HHMD4KhaJJkyYgdPy3334DzbjXHwgbNGiAM4Rer2/WrBnWASs8yB4WgUBwGYjBIhAIL
gMxWAQCwWUgBotAILgMznRLMwwTFxcHfFje3t7t2rUD+/+XLl0Cfr3CwsIePXBoAV5fZbP7pp5/4kvLy
8o4dOzZr1qx6TaxWa1xcHEgRGRAQ0KZNG8G78PX1jYYmAc4pX1/fgQMH8iUGg+HUqVOCu4lKpbJTp05
gC//BgwfAg2GxWNDtz9q1ax8+fBjdZAUkJibjSibinK7QarXvvvvPOOuLSZqamp165d40usVmtsbKxg9meKot
q0aQOSqebn58fHxUr0i7lptls/vXXX8GucHBwMI5DBCsr68KF5yCnc+vWrUF+3YC9gvj4eOCb0
+v1YEk45N69e6COtFwu79DNAuyeGOTXvBhGqzCfRaJCbIvhKhCyDwsVG8Mzd2ojxL5XbpjxC8bDmf4EDC+
5gxwpickJIAL5XJ5x44dgUMcjamDyqTn/22DHw+rdu3RpNEosLTsAhfuVnCqFr167cASg
+I0aMAM2naNGmSnZ3N/C8zvvYhFLFkBXPUxnBZxMz6qGQQ+ODGEIePSASEfLUMhWRH7nLvtW6yBKcqRCLohs2PHDtDsxo0bvr
6+6P0CAgMDk5KSwO2vX78eNHmvDzdz9//jwY4qeffpJKpYJDODzAsn37Kz875IsvvsDR5IMPPPPsDpLSYmRi6X8y+USqVo8PPFixfd3d3BEOvWrcOZnYgsIv3Hj
BtoSDGo0Gjt37gwG7dmzZ1lZGf8qq6U6YMAAnHfzq6++An+WsUmRqlHBQUJLidPT8+//vpLUGGHC+zXX39
VqVSg//379eFFPagaQSCTovFMUhXkzL4TkvbmwgkVRS/eVPagaQSCTovFMUhXktqh53rWAzlCc1wLkviqJwhs
DkSaYNB4eThqNYNbfvrPvCX3U4muArjC4Jh6BtMNfhk2bJ4ToR7MrhhZhrGFNhhwvMuWuY7GERCARXg
hgsAoHgMhCDRSAQXIZnGLzKwbKsyWQ4WQCqYL4RetGRCKohQKBfBh
MQwDBjWbzegvc5qmgUNELpejCgN/05PgZglEhFmtVnBrFEWhs4fuOLAsy+UvrX5QbqccZ4cFB7vd7vD
25XI5eLLcOgGacDviTtEEE4cKWywWnF0YhzNst9vRpY65hmmaRtc/2lImk4Fm3F2AlarFWcNo3Cpbg
UVdojNZlMoFGAU9G3Cf9bP3GDdvHnzvffeAwV
```

```
UtK4cWPgdMvMzASZTm02219//SV4azabLTAwENTXbNSoEXDCvig8PT0jIiL4zmm73X779m004/gff/w
BvE5o+G5FRcWFCxeAlzA1NRVHk6ysLPAgHHLz5k2c4y9arbZx48bgwAq6v46JTCZr0aKFl5dX9c3kcj
l6s+7u7o0aNRI8dKLVatu1awfSYScnJ4M60gqF4o033hCsGV4TGjdujGobHh4O1nC9evUw6wNg5XTPz
8/v3r07SDAwcODAnTt3Ch6dOHv27NChQ0Fae4VCAfR7/fXXjx8/Dsphz5s3b/Xq1XyJ3W6vqKgQF5pE
07RSqQRq7N+/v3v37nxhXFzc4MGDgRvbbDbjZESQyWTg8fv5+R0+fBjkIdi6desnn3wCrlWpVIILUSq
V7ty5E5TX5u5L8EHcuXPnzTffFEyR7Ofnd+7cOZCIecuWLRMmTKj+Qi7P9759+/ifOlardcSIEcCbLp
FIUIXRGaYoioub5QstFgvOKTb0BJ9D0ANBDmnevPmRI0cMBgNfqFAowOdrcnJyp06KyH3Or
WFBQ1lWVjZy5EhQSaBbt2779+8HyxiFuy/wmsycOROcL6ldu/apU6dAagrngi5OztkNnrVEIkGftUPE
f8PCrIfhUA9zJXwJsGgcFotFsGoAPgzDgN5sNhuaqJ9l2fLycnHjWivhSyoqKlQqFfg24fBdcjgDAFk
l4k7EPAe4JQEMFvrJyc2wYG92ux2nmUNslYi7FoWmaY1G46xppygKs2yS1WoF6xDz5wWnMBCiBzgkEo
kT7wsTiqIEDW41kD0sAoHgMhCDRSAQXAZisAgEstADBaBQHAZnBn8bLFYVq1aBTJYZ2Vlia7oPWzYs
MaNGws2O3ny5O7du/kStVo9a9as0NBQvvDy5ctff/21YG8NGzbcsmUL2LLduHFjfHw8XxIYGDh37lwQ
rvnjjz8ePXqULykqKpo+fTpI4I3W0FapVLNmzQJRh1evXl2zZg3f12Oz2VauOz2VauXAnCsDFRq9VLliwR9J0
plUo/Pz/B3pRKZVRULL169fjCoKAgJx6w8PX1nT9/PvDNoRQXFy9ZskQwu7xCoZg5cyyYo/3n9+vXVq1
fzHS8URU2bNq158+b8Zl5eXug2+bfffnv69Gm+xGg0orHlR44cefToEV+i0+nmzp1bq1YtQU/++zb4
NUNCAjACX5MSUlZvnw52LAHdcVryPnz59evX8+XyOXy6dOngzDJW7durVq1StCxGxwcPHv2bPCaOAYn
4NBh8PPQoUNZluU3Ky8vBx53fBo3bpybm/uUMZ7/f9asWQN68/DwuuHz5MmgWHR0N/JUKheLEiRM4Q6D
BzxEREdnZ2aAZiK3Hx2Hw86FDh5z4/jds2DArK0vcDKPBz3q9Pj4+GycrKQi8XSp08fcQo7DH5GycrKQi
sko2i12ri4OHAtGvxM0zSa090hkyZNEndfPj4+CQkJOEOIBjP4OSQkJD09XdwQ6KemWq0+ffo0aHbq1
CkcnyAJfiYQCK8gxGARCASXgRgsAoHgMhCDRSAQXIZnnnJZrVYHBwcLRsnVrVsX3WDOysoWmWlUmlo
aCiI1+NibvkSjUbz+PFjFsjXt24J3gXXq/TGZTHfv3zzeULJJTjgJFLV
anZmZCTQpLCwMDw/nB53Z7fZYmM3mu3fv5uXl8YYU+Pj44AMhRqNhx4Am5NDw8PcXMSHB
yMGRDrROx2e1paGs6SUCgU4L4sFktqairtwf4MZ1uv1wG/IeV2A35BhmLS0NJzNgpNq1a+MkF/b39
weR5D4+Pg8ePBDMmu0QzDTfDqlVqxZQOCwsBDNO/9MwXxBtvvvLFnzx7B4Cmpq6vp5eGFwdt2rRu3Tq8
HRwczBcOHjy4VFfEllxcfG4ceNA+fJu3bbqdOnWKBBwBB/9NF
Hv//+u2BvKEajcfz48xKNHjwYPHgwUjoqKmjFjhojeHHL+/PkxY8
bwTwnQNP3111+vWrVKRG80TYPO6588gBxFGjhz52Wef8SUFB
QUffPABqDkwdOhQcErLrAZDJNmzbtwoUL4UteNgB27dtxXLFTp04dP348X5KRkREZGQCNrdgBrkdOhfmrq
888/B/n7pVIpVkL352Cw5HK5H79/9xrkRNZsZ2GhChmUyGhrmrK+FLpFJpaWkpNZsNvv6+vLfWJPVFF
RIe6BMQyD1hDdhGQSCTivrDI5XJUYZPJjwHHcIpjH0Ox5Ol8lIViSKVSlUrxzw
HBEiccUqkU3JfRaHz48/O1K0LPQ4JUC473pFRUWCi50iKLRAjkN0Oh3QpLy8HEz3GQHcCIp2MwGEQvC
bKHRSAQXAZisAgEgstADBaBQHAZiMEiEAgug/hN95s3b86cfPctbnMm0De8BcvC5fotPp
UM9RfHz8sWPH+JKioqMjAxxmtSpU2fMmjEl5zcnK2bt0K4ktpnjwp7syByWRC/cSJiYnz5s3jewn
sdnu7du1APuZyaYDabgatLq9VOmSDhosWLe9u233wqGxHt5eU2YMAPAEExP
5WCV+Cema4pMZDhgzhS+x2+88MGSrpShq1KhRoFxFdXU1vFbwwRi8P1lcqpZ5sgSVWPA+kfGeVvPqGhzG
fszZ86AGS4rK8MpESzK8MpESCXy8ePH4/WzUW5evUqCGzOzMzESWDr7u4+e/ZsdLB9fHHa7/bfffv111v5wtTU
VJyU1k83jCA1qfzcpUuX0tJSGuX7+OpUvfnw5jnpo8LnD0Z4zFRUVPXv2BM06dGhw8mTJr+qd8b4jCA0ISPD
x8RE3J6KRyWQxMTE4TU4LMJ/PnzwRCYlZ8xzg58dEh8fj+PDD4+eLFi4JJI2pIZG
SkuMlkGGbgwIE4Q3CZP54dpaWlXbp0wdFk//79lpIhAILgMxWAQCwWWXAQCtCwW
7ZKUd7e3uJK3Xp6eqJ1CQUCQsLC4EJAh4xEXIzc3F425KSkpAbVqaZr28vICMbFqtRrxoxJsn6+uMp0ZrM5NT
MJaosLAQxM3UBJZlCwoKgD9FoVB4eHjwSlVlYE5qSFg6ry9vZ2Y5tRutxcWFoK4s+LiYl9fX8E
Z9vHxEYyZ5xyR6JJA98gpivLy8gI3azabVQvLy8giI3azabQVVqh8hkMj8/P8F4WJvNlp+fxD5axRqMBHgbMxcktCRB2o1
Ao0BfKw8MD591E9ecURktzArgZFkzNTFGUp6enuBnGB+t9MxgM3333nznbjU7CqVCs2RunDhQnAXtyQ0atXL8FXFXJTg4e0OfOnc6K
zjUajaNHj7558yZf2LVr1wW7Xr2LSY5NxcZs5wyqJub2+jRo8+dO8ex8eX0DQNCuvXBJvNFhUVBaK
1w8PDDx48KGiwZwC7gEA8Pjz179oB0+Gh2cLVavWHDDmA6Y2NjJ06IGQg7wsPDf/nlF8FazWlpac
OGDQNnLn/P//30QXp6fnz9qP1KikpKTqe6uoqPjXv/51+Xj+fJlvrBDhw5btmzx5btmzhx5NSFLV06VIQ+0Q9KhKV
```

```
lbWu+++K1gM3N3dfc+ePa+99lr1zZRK5dq1a8EXkfj4+HHjxmFGO+KAZbBomhbMmf9U5OTkPHjwQLBZ
QSV8iclkQpeXoRJwoeggYZPJlJqaKthMqVTWrl3bWWG9RUVFqFnXaDRhYWEg+LmkpARn6jChKKpu3br
O6s0hWVlZQGFfX9+QkBBnHRSgabp27dqCdyGRSFBDfO/ePZzy6EqlEhQ0eRLoN24PDw+gm06nEzzlx3
05zczMBFNXt25dUICeoih/f38c3VBsNltaWprgF3YvLy8ciyORSNCq96mpqThfk/Ehe1gEAsFlIAaLQ
CC4DMRgEQgEl4EYLAKB4DJgbbqbTKYjR46AfJVhYWG9evXi71nabLaYmBgQdRwYGNi7d29Bn6inp2f/
/v0Fay5KpdLDhw8L7lk6DIhNSUnZuHEjX2GWZZs2bSpu1zkgIEClUgHhhQsX/vrrL75ErVYPHDgQbDA
nJCQA35zNZuvYsWOLFi34wiZNmuBsCfv5+fXt2xfM8O+//3737l3Bay9durRhwbBZnFxcUBisViio6
OvXbtW/YUMwzx8+BAIs7Kytm3bhs4ewGAwDBw4EDjjW7RoASqYSqXSn3/+WVza6Fu3bgk69Z9EbGwsK
HJuNpuHDBkCnEJSqRTMsNVq7dmzZ+fOnfC1q1bg/6lUumAAQNef/1lvjA8PBwcOrHb7SdOnEhJSRFx
CxaLZfDgwYJOUo1Gg+OudUhQUND48ePBeZ2kpCQwJ15eXjivv9tzqPzsMPh5+PDhoBlm5ef09PSQkJC
nm7Mng1/5GRO08nNAQEBSUhJoBrJ3P6nyM4rZbEbD99u3b19SUgJajh071lmz9KJwbuVnTGpS+TksLC
wjIwM0A+lGnF75GT/4GaUmlZ9Fw7Ls0KFDgSak8jOBQHgFIQaLQCC4DMRgEQgEl4EYLAKB4DKIj921W
CxFRUV8N5bJZFIoFMAj5jBkTK1Wg2ZarRb1iFVUIAARqPRqNFoBKM67HZ7WVmZaAeQE2FZtqSkBBTX
xawGaLVaQQpmi8Uil8vB7SuVyuLiYnCzmNFbCoUCxzVjsVjQlLtarRZ4rGw2G1DYITRNazQa8LgrKir
ERZxRFKXT6cQF+qAKcysHPC+aptH1qVKp0KVuNBrBtXa7HTTT6/VotIrFYhFXI5JlWZlMBoZgGKasrA
wE8SiVShCvptPp0JfOVAlfQlEU+qxRuKkT9Dna7XZUYYYevv0PEG6xz586BYw1SqXTy5MkLFy7kN9Pr9
egrMXfu3IkTJ/IlarUaTZu7ZcuW/1Ih2HlypVxRXbXC8TNXycOAqjZB2GM18G4+QQkIcnVRM+X9MzhYebRH
TiaZGVlAU28vLxwSlqwLHv37l2cu4iKisLRxCFkD4tAILgMxCABSXgRgsAoHgMhCDRSAQXAbxm+4B
AQEtW7bkb7LKZDL3l3fhwgWWZ9miRQuwrS4uDg+uDg+Ph64uuRyef/+/fkST09PGj2wYMHYGO7tLQ
UeJcdwjDM+fPnwYail5dX27ZtBZO1l5SUxMfgd3JmqPR+Mcff4
g7OhAcHAzuS6VSeXh4iOiKc4nExMSAqGZ3d3cwhE6nO3PmDHjcDRs2FMzVi4+vr2/fvn0F3Zre3t6//
vqrYMi90Wjs3LkzWE40TQPvp0KhaN++vaBTjKbptm3bAod4Tk7On3/+KZis2SE3b94EmqjV6o4dO4IZ
rlevHngQCoXiypUrgn5ns9mcl5cnQrGnAyfg0GHw8+DBgy0Wi/V/AeHQdrv990nTOp2O/l/27dsHml2
/ft3X1xc0W7ZsGejfZrOhOhQQ3z99dc0AubtSyQScOGbb76JWfnZ398fXLt48WIrBsBtQ/KTg5OOHDoEbcV
j5mWVZnEEdlqr+4osvcBYAwzBoh+iDO0HOr0KGZVDvR5bdq0CfSmJgYEAAmi1YsJgYEBAAmi1YsABHYUwwp+78+fMeH
h6ozoB69eqlpaWBaxcvXgya+fv7o8HOHO3PnBFOkVqtPnz4NeouJiQ2k6Ko6Oho0Nvly5fRjxJcOKooAm
oaGhaPAz+qyzsrKaNm0qePv4L11NKj+L/4YlkUikUimOG5uphC8BR9qe1IyiKJwXgJtlPK0h6IcV/se
XaIWdm+Uac1CapjHP5qFIKhF3rUPQ54XeBU3TLMvirBzR4D8vVBMUlmXR8wQURYELMdcqZ1/QORH9EN
HXxOFSR5+1wwfxoiB7WAQCwWUgBotAILgMxGARCASXAXcPSyqVgS86B6b3W632wl8FqtaL7DjabD
birbDabrBLQIY5XiwunRIdw4n4Ht9kPJOicYCrMBazyJTKZDJ0ThmFkMhl/Q0Emk6FbFbGCzL4tS4dvgg
GIYRXeRSKpWK29hyuHMkeo/MbreDDLxPAlXY4dShzSiKQhenw/6tViuYE5Fk7XBLos+Y04S9
FqhKHzUCHDMPg7M+iO7MWi4WmadDbi9rVwjJYOp1u9erVIJYSLd9osVhA3BvDMCqVSjQUbdi2RegBwVaX
17+RIvL69169YBD2tcXNyAAQME1WvatCnw1xqNxe2ZK3lkZBN0X31Tqkrnh4RJ9aVLL7CyhzMRE7SEfdPtNmZdo7sVXP
4+jcOPGjYHCZrN58+bNixYt4t1393hu1yZjxXRuuP8X3s5oyL06sZ4fJRVvHpuzq8NomgN7vqjSNmYWM1AMyoo7HfRKxGyqPrXIHaBEtGsbFiNIbiEQqJCZ6EHFvGzHp8pBgZfJ0F1KdDQUEH6enVRKV1ewGXXmEGNTdavoZLbnKHY
4+jcOPGjYHCZrN58+bNixYt4gtDQ0MPHDjAf3koigJ5393c3BITE3BITE+fPny8Y7VVVWVobW1t67d++ff/4p
qDCKWq1evHhxeHi4iGsbNmy4aNEiEOzZqFEjEV1xEZdRUVHp6enVN1OpVIsWLQIp0q9cufLvf/+bb7M
kEsm8efPatWvHb1a/fv29e/cKFioUFRVNnz4fdpH9o2bIleNbl5eUrVqwAFYIbNmwItE0DbR1c3Nr1q
zZgQMMHgN1p1qwZaFa3bt09e/YAO75hw4ajR49Wfwtubm7Vj5hbm7R0dHbt2/nS5k5RL8ilGpfwgt6
Tkf0f5FFIc53TFxmNN95syZONd++OGH4ML8/PzmmzZuL06RTp07oosYbIyEjQLCIIiS7GzRDc7o7ZNKk
SeDCwsLCVq1agWb9+/fnvhZVT1xcHHM88BzQ6/Xx8bFAE8xjDW3bti0qKhK8+7vn5+YFr58+fD5p
h5nTTXarVxcXHg2piYGPR3A2ZOd5SkpKSAgAAw7pw5c0Cz7OzsiIgIMCwuYqLoPh8+OGHYFCHOd2XL1
```

```
8Omnl5eV2/fh00A184noqaHGsge1gEAsFlIAaLQCC4DMRgEQgEl4EYLAKB4DKID83Jy8sDOYjNZjMac
qzX6yMiIgSd1nXr1nWYI7F9+/aCmiiVyj/++IMvKSkpQTNwenp6grhZu91++/Zt1HfmRGQyWePGjUGK
5LCwMNCMpukmTZqADWDMKF+dTte2bVuQETwpKQkkYlapVC1btgwNDQ2v5LXXXqtVq5Zz87L27dsX84Q
BDvXr18/KyhJsJjpUhYu4bteuHd/9R9M0ZpXj5ORkoF5GRgZ6RiQ9PR0sztLS0oYNG7q7uwNNQDOJRB
IRESEuV31paemtW7eAW/Px48c41wYFBYGXzmAwAG84FxAOmrEse/v2bZAh2mAwhIeHg9c/Ly8P3Kxar
W7UqJHgwRG3mngJo6OjtQjo4ayOHTtmZmaWCFFaWsowDBjCZDIJXlhSUvLVV18BNTQaDWoi+/TpU1RU
xL8wJyena9euoJlzvYR+fn7Xrl0DCldUVIALWZYtLS0FzcrLy9EAYxSbzYbOyahRo6p0kEgktWvXjoq
KKisrw+nQtTCZTFFRUQEBAdVbLodeQqvVik4djmfWbrdPmzYNXXWoDnK5HDQLCQm5dOkSGHT79u2gmZ
+fH6owJlevXq1VqxboEDUHDr2EZrMZ6GY0GrlTjXwsFgtolpmZ2bFjRzBE586ds7Oz+c2Ki4uHDx8Od
GvXrh1m5Wfx37BsNhtOHnGpVKrT6cR9kisqEWwmkUhwNJHJZDqdDpzDxMwuIBqJRKLVakH9AhSKokR/
2aFpGu2/anV6eXkNHz78H//4R5MmTZ71zb4QFArF4sWLBwwY8O233/7000/5+fn413KLU9y4ZrMZZ9V
ZKuFLVJWAcaVSKeithiH9ZWVl4tL8yysRbIYeTOWyIYBm3Azzz1dyH7FAN/yKQa/gCiZwyGSyTp06zZ
w5880330STiL1KyGSy9u3bv/HGG4MGDVq2bNmFCxec+MuU8FJBNt1fTZRK5ZQpU3bt2vX222+/2taqC
rVa3bt373379k2ZMgWzZhTB5SDfsF5NBg4c2KFDh+d/Av6FExQUtHjx4tq1ay9Zsq9ZsqSwsPBFq0NwMuQb
1qtJ9+7/4bWikOr1f7rX/9avHgxYxaaNCkjZkgtf6+/v36v/v36NED7MZ
NnDixbdu2gtfu37//559/FmwWGBhYr149vqSsrGzGlzPzAM89tvvvwF3b6dOnW
JiYsDbfvbsWRBKXVZWduPGDbBXcuXKFZCsXaVS9e7dG+ywdu3adfTo0YL3dfXq1XXr1j1VAm+VSjgw
IDu3bu/kvvrTwXnZb927Rr3jGw2W0JCAnC6N27c+KOPPPuLPFcMw33zzzbvBr1/jNQkNDZ8yYAVz7kydP
3rhxI1/i4+MzZZ84ccoF7h2LFj0dHRfIlCoXjnnAM19fX3CKhWGYmzdyc/PP3HiBNjsHzlyIzI8
ePfgSk8l07do1wXQdWq02Kiqqdu3afOGFCxc2bdrEl9HstK9hstlwNnTohOGjHv7+/j179gyBe7Pz7
feegstEe8AHFeiw5zuNWHv3r044748wc8oDlOkY4IGPzsEzelePTKZbNq0aTjK/w0pKSlBz/S9/fbbZ
rOZ38xqtaJlRFq1alVYWAg6nEmgWFhaWkZEBmi1Zsg
Tn2aHBz0ajsVOnTvhPX5A1a9aAIVJSUgID
AwUvJMHPhGdChw4dZs6c+bf9JUj4O/B3/+HwyuDl5RUFRUFIR/SVUauYqKCpvN5jDD38tAVTo6hUI
hkUjkcvlTffEkvEoQg/UqmQEGD3nzzTZGLMump99+y8zMNJvN5eXlXl9lgyeVypV
KpUqmCgoK6ewdOnSuXPn2rVr41r69rVr41uumkTwEF4eiMF6D39//ggw/QgC+UgoKC6bL37
t0NGzacNGzcAcMGHCjjTYFEUVatWLZzXBjMewsfHByeZpEwmu3ml9SWlrq5+cHrjUajampCh
SFFUUFAQiInx9vZOSkoSdFhkZGGTUqVNH8G2x2WccU2+sLCwEjskKkiogYNGgh6CSsqKrp169ayZcv
qm9nt9jt37t37ixcuPDYsWP4kBJK32+25ubmCKwMCwWa
Q5goKCwMrx9fVNTk4G60
ShUIBmfn5+q+pampwK/Hsixok+ubm8cXFYEmUVwJ0U6vVtWrVEmd8LRYgChnJyckAS8jAUFB
bm5uYK96XQ6cF/cN3cwyWq1OigoCAT2ig6HcrLBksvlX375JY5rAzMGfdKkSTju/z179oBfQ3q9fvPm
zSARbUxMzD/+8Q/rs8JAm/+Z/H/+S7lLdZ+TtTMiXunIYR2nSX1WlyV5BLlVf/G6sFQvQbK0LTnlvW
+u9BtuXXp0uXUqVOCv3quXr3KTX71ze7cuTNhwoT4+Hhxhc5fQioqKn788fcHHjx9v2rSp+uzyFEUlJS
U9ePBAsE+appcuXQqqENy8efD4W+fffZZbGwsX/Lo0aOxY8eCt33du7uGg/+/du8cX/
vrrr2AN2+12NPdJy5Ytd+3ahRk7QoRdGi7Js2TIwRGBg4M6dO319ffnCb33vRb99xxGRk
fHw8X9i8efPdu3eDj39xKSg4nPwNy8PDA83DLRpdFIpSKVitVr1ej3QxN3dnaL+59yZQ4W1Wm1
eXh4oJYDi7e3t5eUFHjYKwzDokaiKSoRuy81kMvn6+gqeqAoiBHCHl5ub28cffyzsrgtFf/v9/Sh4
hQ/YdHRERECcHAwOClHl5ub28OHD/Px8LSwOx+8kBz/PxkJIpO0CHlLEUOcI
a+vL9Z5JQSGYdCMQ56enkA9zK8/ykr4krKyMvRTUy6X+/Ov6OjE4jBxrcHm0Wq2Xl1c1DRiG2bNnz5Ej
R14xa8XBsuyxY8f27dtX/d15e3u3u3bt36OepFeCYQg+Xy6HS66n8PPnr0aPPmzZg7OK5IeXn5t99+m5G
RUU0buVxODNYrADFYLo9gXq5zDg4ODsOWzZZ8+C3LCvHgkJCZcuXaq+Xaq+DVrSkeByEIPl9hbcv
P50qVL1f8qrFu37nPUiV6lUkydPrjpJFZfXtLSVPRNbhJB8Z+aM6nPUOPP4LtTz8/v7Fjx
4IvCydOnAC5n3U63bhx4cq5f6+fYLMGDRrRgFHCv3o1osViq
/630ynD37l2WZauZMXd39y++aL+vIGz30Y+aGvbKOoLjjTmZ9fPVLIYgzrS7b3sXIsBX4eMD3aJv3kjhF3ZbZXBn7fS1
4c4BmDdZoNFOnTsU5OoASFhYGNuxZlv3hhx8SExPb5woCAgDFjxoADRr179wbVmw0GA7rFg3qFDBzBLKp
UK9ercuHHjwIEDfInFYklJSRxU0+H6ChEFIeVn7t06VJaWgpaDh8+HDTDrPwcGBiYYkpICmq1Zs0bcv
SsUihMnToDeMYGWnMAMfnYKpaWlzo2hfWnp3LlzWVnU9O0zWY/Vo/TSj/P2sYhhk4
cCDQpHnz5vn5+c90XMzg527dupWXlztXhxxzXZTdefstXPKT8BXHZDK5+jFRTMrLy9Wr
```

```
9m9zp3xlisF5xuKjmF63F84AYrL8DxGC94nC//F+0Fs8Drrzji9aC8GwRH5pjMplArJPJZNJoNMChoN
FocnJygOtdoVCAZgaDITc3F3xCUhQFmnl6eubn54OYAMw0sgqFAkRdKBQKnBJsnHpoPFReXp7gG5KXl
4dG/0ilUpBJlgsSElTGbrcXFhaCuFl/f3+cWyBwrj3+nwUFBTjW3GKxcBlH+UJ0Z9Bms+Xk5AA3pUaj
AZ44hmEKCgpAzUGlUomGBInGarVmZ2eDdSK6PCjLsgUFBeDdROMcnw/iDdbvv//+0Ucf8SVyuXzhwoW
rVq3iC69fv96vXz8QTRoVFfXZZ5/xJSkpKZGRkcD0/OMf/zh37hxfkpeX99FHH4GQKMy569Sp0zfffM
NfTxRFgaMPT+Kzzz4DvpiHDx+OHj1asMy9p6fnl19+CUY5fvw48NzpdLpvv/1WMA+11WqdOXPm2bNn+
cL79+/j3AKBWwP8P8vLy3F+RSYmJn7wwQdGo5EvBIuQqwU/ePBgcMpk0qRJ06dP50uKiopGjx6dlJTE
Fw4YMGDlypVPcyvVce/evb59+wLTOW/evDFjxojorai1oaOzYseDsMZiN54Z4g1VaWgomXalUenp61q9
fny9MT09PSUkBH0darRY0q6io4EJM+UKapkEztVqdkZEBqktgwg2Kc7IJxd/fH2hisVjS0tIET9MEBA
QEBQWBI4sKhQJMncFgADb9STx+/BhcS8BH3NSZzeYHDx4IfjjZbDb0IBJaiZphmLS0NKCJ4CHHp4JTG
AgF9X8SLMuiCr8oyB4WgUBwGYjBIhAILgMxWAQCwWUgBotAILgMzsw4arPZjh8/Dnb77t69izpiTp8+
DRLOpqenoxkX//zzzy1btvAlRUVFOO4JuVyOhnp6eHhs3bpVMB/2vXv3cE5auru7R0ZGCirjML4Uk9T
U1FOnTvG96Vw6cHG9EbiExYJt7t69C1ZdaWnp8OHDwUGEuLg4kCJdp9P169cPhBxLpVLQm9FoLCwsFK
d/enr6yZMn+ZrY7fbU1FSca+Pj48Vl/uQyqQo2o2n6rbfeApVZ9Xr9zp07BT1dXl5evXv3dlrlZ4fs3
79fuPcXhIeHx+XLl4HC0dHRokun7Ny5s2Yxm//D+vXrQf8Gg+HixYugGWbl5+rHSktLqz7f+StDeHh4
WlpaDR/Ny1P52SEnTpwQl9D9OaBWq0+fPg0UPnXqFI4Zatq0aV5eHs4MkJ+EBALBZSAGi0AguAykkCr
hBSOXujUI+L//d3NzKyp3e+DME5SEVw1isAgvGF+927YpbrUq6/78fNVt4mY3hoQwE56Akw2WWq0WrK
PnMIZLIpFoNBrRJW2Bh5HLtAliESwWi16vF5e6wGazgd44hQXdHyzLlpWVgRhpzDxzUqlUr9cD5xRm+
JsLIZG4uWvcPLX/999ap9WvE49KpdLr9XyJuJhhDrlcDnpzCE3TYIFNIW+TTRN6/V6wSguhmHQwppK
pRIz1B9gt9vRNfykZuAuzGYZWtWpJmvYyZWfly1bJlgz3c3c3a9N577z2cE0TTk0Oorb2Y6mOiosArXK9evQMHDvDzUnt9nnz5uFU
PneIMw2WRCKJi6ovQor7atayZUtUjBIUtRUtjgNa34atkZGZSKSepxPCav
Veu3aNZwi6Sienp5t27blSywWixOTkBBQKIp67bXXnNhhUCWCzZKSksAC02g06Ck/d3f3Nm3aCPYml8
vRHzqhoaE47yZKXl4eOFzmEJZlQS0M7r1u1aoV//yX3W6vvu5v9RAvIYFAcBnIpvvfBYqi3nnnnV69e
qH/xLLs48ePP369aqUZBqNZvz48dHR0fz48ddOXQ7tUCjsdvv169f9f37t3LMMyg
QYPQwlbcEI8ePbpw4cLNmzdBkINDAgbi4OMFmhFcDYrD+LkgkkrZt24Kci1XY7fa
SkpJff/118eLFN27c4LYkPDw8pkyZIpVKLRZLUnYZLSUnJV199dbVq1ZpaGiqhuYmEWLFt27d696bb29vceNG4eTXvH+/fvEYP19ID8J/0ZQK0
lU2rVr18mTJz9piKKiopMnT86ZM+fIkSPHjx8HRSIdwWPw8O4yIyM3OHZbDt37rx//z5FUQqFYs6cOWPGjFm0aNHdu3enfp2/jU81/M+8dixlJ9Ti
YgVamBNfVcAznjPCy4X4b1gBAQF9+vThS2/vAnJO2vvVCOcIQwfYFx1gTkqSiVyvbt2+NsAaJ
kZWUdP35c3BJPT08XcdWTCA4Obty4MV8il8sTEhLQrLsAhmHq1KDp0LFovl7NmzXESrRCLx9vZu2r
Sph4cHRVGdO3ceMGDApk2b7HZ7SkrK7t27Fy1aRNP066+/7t9vtXr+3bt9vtdqlUOmjQjQoNq1a7u5uZWVl
X377bdoXk2z2RwbG8tlr+Zy87/xxht4ikfSqZZPfu3dUcMSkqjpy5EhV8fH5+OHT+tyzqbbt2/f
unWr6tqnyoR57Ngx4/EH4e/vjxPx+bmgt63Cd1vbBP/L2ZXWQeEfJ0Kh3A/3Xj4z/aVlIs/3agMgCmkr4a6IBJ8GBgyCD6ZDgoKCTpw4XY/97XZ7Zmam4IVPBC
fg0CEMw1j+l+Li4h49ekj/l+7du6VSqdLSUuhI16X7U2dscoZJ2zx46Se2kNg8nvU2PDqsnA2qBQLIP+hxb2xgR27a7a6iv1c4AxI3CZqxUFFXKv6IsTT+n/xGwM+q
Hx8fzledmJjICWNiYqp8uzjyCWfTWqrMtrpz7CmVJXcIU9kfgJRkcL0FE1iGkDhlJI8DZSpGFsjqJx4/jCXJJ+xdw7iMOz2uiypDYzTTq
7f//7/hM32tGX/s6dOxcVFXHXcgYUbYMT/Azms0+fPmazmd/AarUOGjQIZ0msW7cyqrOCSyM
7Obty4Mejt/ffftyAwDCPYG8MwQ4YMAb05nJ81a9aAa1NSUoKDg8G8zc3PB56ubm9t7772HK
oxy4sQJrVYr+DY9j+BniUQiva/gHOPTwXaW03OTNrtdrQ3TJxbKQtz6hyCXltDZbjdq2PHjqWl
pXESPz+/KmuSkpybds27qRi69ate/bs6eNN+Ly8PMFRioueLDx48WHUezcfHBycyrdHOPSBvVF
BWZwMw+A8CIqiwINw+Co6BB3CbrejSwLzCA7aG/5Lh94XZqk0h2sYhaZpdIiavE1kD4vgxp2NrPpWVV
BQUJURjGXZAwcOJCQkcEfAR4wYUa9evVGjRnRGG8ty5c7/99hvmECqVqunhIakhSBAB8RL+HaFpu
latWlUbTwaDYeLEidyeyeldNlsPnPmDH8jMiMjY3bt1+wYAH3H8PmTH1SYUyPU2aF9sSmEolUpr
```

```
165d9a/u7u6TJk3isrtVVFScPn0aZ5OFQAAQg/V3xGAwrF27ltttpShKo9F4eXlx395jYmIOHjzI/9J
ut9t/+umn0aNHN2/e3NPTc8SIEdwXpdjY2DNnzjxpCA8Pjw0bNlRFvWm1Wm4Iq9V68OBBvleRQMDnmR
sslmWtlQAhzrUSiQS4F/G9jZzHnS/hdijFNZNUAi7E1wRVG0yI1WrFeYG5bThwrYhtLIlEAlLZcty7d
2/p0qWojyw9PX3nzp2NGjWqivkoKSnZtm1bNSVsJRKJw5CUhISE5cuXo17F5wN4EBRFWa1W/gLgNqdw
lhm3Qw96Q5cTwzBgteM/a27TXXAIdIE5XMMO4XbB+RLMjTPuvQZCmqZxLnd4CziDPg+DlZCQMGLECKD
QX3/9hXPtmDFjwLFppVLp6+sreKFGo1m8eHG9evX4wvPnzy9dupQvkUqlCxYsNF7yauV/4hcZjRQxfojjjz
/+uGvXrnyJJVqvFO0Yft6em5ceNGUEf21q1bgwYN4kusViOAWeGYRYtWrRp0ya+ELjqMbFYLFUvkkQi4
XbZ69atO3fu3GnTpoHoHM8SzLHjp0KDIysiqyPTY2VjB+1eEQr7322uzZs6OiokBA+PPhp59+4v+ZlpY2
YsQIvkGhKKpXr144qd/Pnj0/LNqxHQ/v//y5ctBoNzBgwdd37tzFr2rQB0Sp6vV4qlQKD9cYbb
/Tt21eEJkqlslu3bkD46NGj48NGj48ePi+jNbrejkd4iMBqNK1asqDrjptPp+vfv/9ZbbykUin79+e3fB
ggXg8zMtLe3w4cPNmjWjadpkMn3//ffVfL3i3ILLli2rqtFtMBiGDBnStWtXlUr13nvvJSYmLlu2rCY
eZHGAh/jzzz//8ssv/DulaXrcuHE4z/p4JXxJWFgYmvglKSlJ3LNmGOOb8+fPnzp3jC/mfARwURaH5Ua
5cuSKXywUNllqt7tGjhwduArk4CNHrVZPnToV59oWLVqIe5vIHtbfFLPZfPLkyT//LNKcuLEiQMHD
rRs2VIqlfbq1Wv16tXghyF3gIhhGJqmGYYRPPtnMpl++ww8/vPHGGzwkoaIwtnFhs3
bvx/PwyrPkTIvhahWsixBoIbFwbAnV3gfinUJAHIk3j06FFV+hFvb++qtHalJrcDF9y++/3//nf6lhv
ZiydUA/mGRXDjdv1CQkK4/+aOsDp9CK1WW6tWLe6/uQLE3H8Xlp9ts/poxFeTYjB+jsilUobNGhQdR
JKpVK9++67VZnhHj58iHN4vXpkMlnDhr2r/lSr1cYmJiAWPz8vvLq/2hd/j9G4Hrz8iWUlJ
ARKapsPDw0Ee3oKCAsE0LOIwGAxr1qypUl6hUHydz21uHDh2tuTTw8PNavX181hFKp1Ol03BAmk+n
gwYOCW8LPgvPnz/P/zMzMbNWqFX8bWyKRVEcV1FaWpqYmAi+darVapGaPTAmtAawHPqaioYApPi4mLg5q56/fkSbhWJ6P+pwDJYRqNx8uTJVXscHAMGD
0sFkvt2rXBtQEBAZZcuXRI8deGw3HpqaioYApPi4mLg5q56/fkSbhWJ6P+pwDJYRqNx8uTJVXscHAMGD
NixY4fgtU2aNm/fz/Ir/DPf/7z4MGDfElKSsp7770nugiFi0s37lzw+G02G1o0H+Xhw4fDhg0T
900EVVij0axZswa4NWNiYiIjI59FCAtFUQ4zAVgslkOHDu3atavmgz5pCLPZvG91//PGFhOaAvOl
dunQ5fPgwsDJoLpC7d+8OHjy4KrUhx5IlSxYsWMCXcHVJwLUjRowA7n+HREdA93UavWOHTtAQuSzZ8
8OHjxYcME7LEKxYcOGbdu2CWqCwlWXAMIuXbqA3riqGSL6fyqwDBanMXhgT1X6BTxI9IOIZdknBXk8U
9CPDkw4hcGciIaiKPRgl1qtdm6+J5PJ9KSvTmazM2m4uKiuRyeTVVTyoqKp40
hMlkun///oEDB/bs2fO00B0e5qkXckRScj5MnAZ4XV0hJsJYMwzDos6ZpGucHgaISwWZSqRT0zzCMUqk
EQygUCqPRKFg1xyGmSkRc6BCZTCbu91ANIXtYfxcYhtmzZw8418PB5QJ9/PhxQUFNYewT548mZKSQl
EUy7JojQ/OrGzcuPHIsW1hY+Pjx48LCQhHK3759e+TIkdzGQlpa2vM/wEV4SSAG629EaiWiL
8+qpJoGdrs9qRLRQzyJoqKiP/74w8DMRgEQgElwHrJ6FGo1mwYAHYKA0NDQXNZDLZ
p59+CtIfl5SUfPrpp8Al1LRpUzTCTCDoeioqJly5Zxycir6NatGxiUK6CwceNGvjA1NRpk0Sr5evXrTpk0
DG6U4nmk3N7eBAwf27t0bpyVALpdzBbUIzxRCTCCZPnsxP7lxDYmJiQMS1Q9DQd7
PZ/N///nfXrl18YXp6Os5JmuDg4E8++WTgwIF0SG/8C30CW/SquKH1HSVO8r
5cv1Ov1q1ev5qdw4OJhT548yW9yW9Wq1at6dOnY5XXxsmjXBNOnz6NOjv37t0rrrf09PSqA9lVfPjhh6BZ
fn5+8+bNhW/eEZ06daruVPKnuWLBF3Xw45dOgQTtqN6jupyun+yoOT0x0QExMDHNY0TR8
6dAg0u3jxIuoRW79+Pc4QSy5Yseb7T8H80b948Pz8faPLhhx+K683+v69euCd1paWtqlSxdwbbdu3c
rLy/nNWJYdOnQoaPY8croTCATCc4YYLAKB4DUQgAwDzI0ISaxBQ9kr3KJ2dyPe5gX9N2l8+
SNZWVk4GYFR8vLygoKCwEapTCYDvRmNRlL9fX5AiGZOogCCm5LL5aGhNIoUOp5S0aGhoaAWkBhY1xb
HDcwMPA5RGWR+CgUirCwMBA6hm7DDWyyW9PR0cAPUp9B9B8vFcFV6cYJqAgAgEKxduDFRcXP/4Yfz
rVi3BUGq9Xp+VlVVBYBATwJRRFBQUFgZcO823CB8tgFRcXf/DBBYByC9dL9+/zMl6EvuSbTt+f
LlIi708fH57rvvQHWDvXv3urUDpeNXCuQGTW6ugLK0pBgJAcUQlaYAbK34M6eiK6B+mXz8mRKGDRtWl
Sn4Sej1+h9Eew9+vEKEgUytmUoevXp9803CtRbXF4oF/1//HJ56aX8ACUgPpCT5ySBU1Bd6V4KOCNyl1eVT9/b1KDaWPA7VWWI6IeX28Ljzr+p7c0
4MJRo0ZdvHhR8NrFixeDwcB6+ysHR0cDuoBQWFk6YMAF8Jxg4cCA4NqRUKr/++msQmC2Xy3F
CKfHBDX7Oz88HT6ImGUiKKxFxIU3TXl5e/v7+fKFUKgW6WSwg8EAmjkXmUyGUw6DZm8vDy0DZ
LhnK/hUiA80/sioMjlcvD1xyEMw+TmoCrwWEtLSwjK4eHhTt6+srek1UKi0uL
```

```
gb3hb6/EonEYX4O50L2sF5xqEpetBbPg7/Pnf6dIQbrFUdWyYvW4nnw97nTvzPEYL3ioJUyX1Xkcjkx
WK88xGC94iiVyr+JV1GlUuG4WQkuDdZnr0qlmjBhAsiFhOmDCwkJmT17NthRjo6OBj5Hh3Tv3r19+/Z
8iU6nw8lzWFFRsWXLFsECllKpdNiwYSCM8+HDh/v27RPMQuvj4zN69Gg0JS5Aq9VOnToVZK27fPnyzz
//LHQTzkGhUHh7ez+fsV4sXl5egrvU//73v/l/3r9//yXJBXjjxg2gG5eGGI3OQ+EX++B4/Pjxnj17Q
E5gfhnKp0KlUk2cOBFsujdq1Ag0s1gs33//PUi4FhoaOnz4cGd+88UJOHQ6ILnCk1i+fDlOb2vWrBF3
7wqF4sSJE6C32NhYQTPE2evs7Gxxt79+/XrQm8Fgu/XrQm8FguHjxImiGBj/LZLKYmJinGothmNmzZ4ubH9ciKiq
KYZhqpgInn7XD4GdMkpKSBI8I4KPRaGJjY8VpcvnyZfRIIEpISEh6erq4IVBI8DPBCUgkklatWjn3LM
xLiEKhaNOmTfWnJR89evQcNSI8E4jBevVp1aoV+gX+FaNBgwZVRQ+fxLPI3Ux4zhCD9eoTFBT0wQcfv
MJb72q1evz48SD+AQVn25TwkkMM1quPRCIZMWJE3759ceLLXA6JRNK/f//IyMjqkx3a7fbExMTnqBfh
meDMEzpctSicsqZyuRwzrgX4Jrjj/2BpqlQqnN4cqoET/cAFjgGXucFg4JI6Vn8hV0nYWSeh7HZ7YWE
hmBOKotzd3au/EU9Pz4ULFz5+/Dg+Pv6FVDB9Rkgkg4dOnz++eeCe8xcAL+4UaxWa2FhIXjWOp0O50
urWq0GEaZcxTNNeSftdntxcTFITFBSUuLl5Qv8c0ajEbOWKIBTGFSiRGsmurm5ubn7gzdRrVbn5OTw3
x273a5QKEAzT09PzCgFZxoss9n8ySefXLhwQbDlp59/+OZ1+OmfOHMFmO3bsePPNN/kSX1/fvXv3gvcO
BcHPmFAUBbp6EnPnzh2h2eyKOjbvzGgxJFhnWGEBgkYQEIpoBBCJIaBRnpMmhe75zmO/49as7QUSiRGsmurm5ubu7gzdRrVbn5OTw3
```

```
U2FQpFUlJSVVUVV3j37l0QXpdIJJ06dQJnAlq1agUWpWm6U6dOvD19KYpCQxPV1dXnz58HsTmlUgleC
avVmpOTA/L87927B75Nq9XaqlUrEOmuV68ezid8+fJlNLr66quvRkZGciVxcXG43aFxdua9Rgl/qwP5
G0FBQYcPH7ZjMHr0aAkGS5YsAQPv3buHUx9dq9WePXuW14zq6urevXvzXpdEIkEPUjVr1uzRo0e8S9y
/f79JkyY4F4viNeImFotxxqanpwNLrl69Gh0dzXtddYkSMgwDFjWZTAMGDMC5w+np6WA2lmXR+4kmde
t0upYtW4LZMCv89e3b12g0giXQDPlz586FhYWBJdasWYNj8Pz588GiYWFhly5d4r0ulmUdDgfvC+bV4
AsXLtSrVw8YvHz5cjDw9u3baFhTJBKBgZGRkRcuXABjf/jhB5z30Os7vHXrVjCbw+EARw6ehvBfWOgB
MPoJOL+wMA+PURQFZpNKpTjdNTyfBK8lLMuiX+wz2ca7hEQicblcPjxljpa094pIJAK2+dwSnEUpisK
8w+hxM/TpPw3nEwQbzLuKx2D0HCZQ82ow+sVivpweNb4reAaDMb8m9glcidPpFIvFYKxIJBL8LqGz4U
P2sAgEgt9AHBaBQPAbiMMiEAh+g4/7EjIMAzZZPLsYOBtP6Eap2+0GszEMIxKJeLdUvSqg/zl3uVyY5
XHRSxCJROjFong12LO/yLtEXfB663Au1utDfHna6qAPEfO6vOK5S7yp6Z69TvAQMR8WRVFgoICWPwLw
bChzJZivBLrt6NVg9Lq84okePKPtteFLh+V0OufNm7d69WqusGXLlnPmzOHNJ2rYsOHChQtBFuv58+e
HDRvGlSgUigULFvCmiUul0kaNGgFhbm7ukiVLuBKGYa5cuVL7VB4++uijnj17ciV6vX7atGm8vUgVCs
XcuXPBdf38889r1qzhSpRK5fz589GYvWBOnjwJbp3RaAStQL1SXV394Ycfgjv84MEDXxlWRzIyMrZt2
8aV2O12wXUKL126NHLkSF6HFRYWtnbtWrBPjB5E8MqwYcNAQWSZTAYOcPicRo0abdq0CVQ6ycnJAa+E
xWIBhzA8lcTHjx/PlUilUrQuedeuXXft2sVrydWrVxcsWODLUA9OKNHrsQZMevbsaTKZwISjRo0Cal5
ruqPJz9HR0YWFhTg2o+zevVvwT5hNmzaB2a5duxYeHs470Gvyc1paGlDTaDS5ubnCrssraPKzb/F6rA
HF4XAMGjQIZ8J169bhXBd6SuB3oEOHDnq93heP5UUiOPm5LnhNfs7IyBA8IdnDIhAIfgNxWAQCwW8gD
otAIPgNxGERCAS/4bn3B66pqblw4QLYeNNqtaBvaKNGjdAkhvr16wM1jUZzU79y5gxP/DgwM
bNKkCQjZhoaG4iwRExMTHR0NBspkMt6BLpfrxo0bQBgcHNy4cWMfnnVAiY6O5i3f7BW5XP7w4UNwl7R
abePGjblBN4qi4uPjcXrEmkymc+fOgSWaNm0KYnPR0dFgNqfTeevWLZAPHxYW9r7311MPK5+cL69
VoMpkuXryIU8IcB4lE0rRpU95uAB6DQSV1jUYTTHx+P0yIAJTY2FudB1NJXvHYYhrlz5w6oEP348eM2b
dqAmu5VVVXXgWSuVyqZNm+KcHTNm+KUq6VwvuHKysrevXtgtjjjyAA0DSN5cuX+7HU3aXQH2gucjBOtAe
YcOEhbGdTidow6w9vTfbl/Y/fj0pVV//1r3/xKvVQwONBZFLz/2+mbd1/7ilexD/9m4MMFPfbr18+TE/ubxO12m81mnP
romoDBgzYvHmzsKbzf/3rX//1r3/xqn3xxReff/65gPmNRuO4ceMuXbrEFfbr18+TE/ubxO12m81mnP
zHWbNmbd++nSuJi4s7dOgQKEVgewJXUlZWNmjQIFB24u233wbHa0wm07BhPBw8B3gonn5fTVX46wsLADB
w40a9asdjWTyTR8+PCzZ89yhUlJSdu2bcNq6Y5gsVjAQQevyGQmfqFbPZPHLkyNOnT3OFr7322oYN
G7gGu93uDz/88ODBg1y1li1b7t+/H0Pi6PRCLRarW8YxVP4ErMT8A5TyQYsVis0Wi
E/TlVKBQ41+UV9I//79BPUC6XCzOYoiibzQYeBKjx8pu/xpkQfU88lfVRg8Hnarfb0ROMUqkUXBdN08
Jcv8c23tYS+EilUsyzlOirjt5hfJRPEDwcB9Rgu92u0Wi4X7EnMRuoeU4+4SxB9rAIBILfQBwgUDwG
4jDIhAIfgNxWAQCwW8Qvunepk2bSZMMcUMnDocjLS1NWP3yoqQKi5cuX8+4pms1mUIiYQD
X76MlvpFKSgomD59m+xR5lMNnv2bKDWvWVvatW8eYYu1yuyNWv
WXL9+nfcqGjduBJbQ6XTFTc/ffr0li1buBKWZX/44QeQMc2yrFQqxSIQBNxM5kZF4TN5pWzZ89Onj
yZK3E6nXfv3/9lavWsGHDqVOn+mrH2ul0rl+/Hq+/njdtXiQSDRkyZPDgwbxLs3Ts57XhwxgRZObBxyvrMIXDRkyZPZo0
Vxh/fr1cUIHDx48WLlyJpSYDBJpNp+fLLLtIDTfvn37dFMzaxVhRiOVxY4HuBsKgHN6
hjpOc/ingpNw+Dt0fr5y5QrmWSeUqVOn4lxFXZKxSFXZKfUZo3b15aWsq7aHFxcVxcHBibkpIC1PR6PXGJjk
5GZT6ttvtON3GAwICNmzYAOzYAJa4efMf2l5h/vz5OLc0x9EHBAQMGjTI4XDwc3bH8Tjc0tKSnhn0+
l0CQkJOOah9O/f3263gc2dzOp3gwI3Pk59LS0tB/QavqFSydOJ6iLy8vOWDgYTLhixQphs5HOzwQCg
SAE4rAIBILfQBwWgUDwG4jDIhAIfoMvU3MoikLkL3mtyqgSqpVVVSSdVFSXdwHGKoFXV7/fz7fSkULZaWsP8BunTnPdtbeHw9Zd8/UqHW1hA6G
BPhwiIyOFRX/q1av36nlnaLOAS20c5USAcqqKvIiXBTSt2QxUTi4d63BwFD4Ko9gcHNzqH3nexgKJmq
YKnEJRMsa5+/Bg8RKVZkSGRERIZDWPVfJHMTfd3UjqvpgYOVZEXo9jhFkqXQUgf7qVAQ3JFOreUUjX
G1AQMCXX375t7/9jS1XFZD04sCAgImY54MkoR1Ol1ycrLgGt4AqVS6bbNmy6HWDNh1y6cePGfh77969cePGfh
Yenp6SADfufOnd26deNKWJZPq2ujCSSb775Buc0z58/f+HChVxJw4N/+d//gfkl4Ji809jyJAh3bt
351U7depU79690T88AIfDUZecOEBwcPDWrVvV/fjjz9+sknXIlCoUhLSwOh2Ozs7D59+oBaArzP1M
M333yzefNmYQavXbsWf87/d+6cNm0aV2Kz2QaPHgwUj/1LWPIzik6nGzp0aaGlpae1qGo1mz
Zo1oOHHA4cOHwtVsFU9X5zpb+r/4+BcW5tEEu91eULLCq6ZWq0G1FrfbLTiFFcVjMFgCE71eX/4EXk1
CYlEgnP5KPh32OFwcKNIbeAwjkBc4jZgk4ajqdzoe/f3GgaRo9roH+chSJRGhYeEBBeAwW1iKhpqZ
```

```
G2EN0uVzBwcHAEvQUgtvtRo8c4vxVw4dhmLKyMt6rcDgcqMFyuVzY5dcRsodFIBD8BuKwCASC30AcFo
FA8BuIwyIQCH4D1ga2QqGYOHGiTqfjCmmaXrx4Me/Yhg0bDh8+HAQ73njjDVByu6ysbOvWrVarlSs8f
vw42MGtqal5ruVGAwICHj58mJGRwRvaKCsrE1wRtE2bNl988QWvGk3TS5cu5VWrX7/+yJEjQeX4QYMG
RUVFcSVhYWHoAY6TJ0+CmraYyGSykSNHguhn48aNZ86cyRvGLi4u3r59O7jDAwYMePXVV7mS0NBQnJ1
+s9n8ww8/gP1pp9MJ7rBUKkWrlcfFxX3++ee8BotEotWrV4NE1PDwcLCEXq/fvHkzCIB26dIF9AwPCA
jYs2fPvn37uJLLly8DHYlEMmLEiNjYWK6wcePGaAP6Xbt2gYoDDMOkpKSACauqqhYtWsSVsCwLmjx7G
kT/8ssvAT7C8/nzlhKIjIzErcssLB/S7XZnZGTgzO+18zPKC0l+lslkhw8fBmonTpzA+U4w8dr5GZO9
e/fynmnylM2uqakRtsScOXOEXRdm52eveE1+xuz8jOI1+XnChAnCZvNKbm6uRqMBS6SlpQG1goIC8Ef
CUzUfqPk2+ZlhmKFDh4Kxbdu2raysBJpo5+eGDRsWFRUBtb///e9ALTQ09MqVK0Bt69atvJcQEBDQq1
cvi8WCfaf5If8lJBAIfgNxWAQCwW8gDotAIPgNxGERCAS/AStKyLKswWAAATuv0TqNRgMCguhupSfYB
5I/q6qq0GCNSqXCyUymaRqnF3R1dTUo1ep2uw0GAxhrMplCQ0OFZUSbzWac0KHVauXNmvaohYWF8caw
tFrtc+0O7cnDAFmHKpXKZDLx3naKotRqNU7va6PRiPMQMZOEMLHb7Z54BVeo0WiAwWKxOCwsDLzYLpc
LGGwwGIKDg8FnQlEUUKuoqECzl2QyGQhEKBQK9A5LpVKNRsP7uJ1OJ7oKTdPh4eFciVar1ev1IITHsi
yqVlNTAyzBTPx2OBzl5eVozrngdEgsh1VTUzNp0qTbt28DIVCTSqXffvtt586duUKVSoUGLL/++utDh
w5xJTabDZ3w/fffnzhxIq95Bw8exClWjfoIp9M5Y8YMUMCgWbNmu3fvFnZDly9fnp6ezqu2b9++BQsW
8Kq1b9/+yJEjvIFCpVIprFUvPoMGDZo/fz5XYrFYUlNT79y5U/tAmqZXrlyJxvVRvvnmm3/+85+8aik
pKWi0SzDnzp2bMmUK98Omax5uWg0nfTpk0PHDgA/nKsX78evHURERFr1qwB6eW7d+8Gai6XCy3e36
dPnyVLlnAlNptt8eLFM2bM4Apfe+217777jvcPwJ07d5KTk8GBHBeoaysLCUlBSR7JycnA7Wam
pqvv/764cOHXCHm6aJLly71798feNhly5b169O9bevXs3btyoXU0kEv3pT39q0aIF74TF
xcW8s3neAJzZjh07hjMbitvtBo/BE8RNSEgQ1vkZ/Gl6Gnq9Hsfgxo0bN2/e3IfJ3oIJDg4GD6K6urq
0tJT3KsRiMWbbi5In8KrhZJvjYzKZbt68yU1+pmka/cOmVCrRzvJ2ux1cs1ma9SoEUgS3rdvvH86z1m
q14A57fl6BsTExMTgdkm02G/h54fkRB5ZQqVSFhYWPHz/mCgcNGTUKioqMD9YFLPZfPPmTSCsSxtts
odFIBD8BuKwCASC30AcFoFA8BuIwyIQCH4D1oauVCodNGgQaKb44MGDU6dOcSUMwxw9e5RvzOBwW9ehTs4Xnl/v37
QKLVavv06QPCHzabDVShSgU/fv3590Rl0qlr7/+Oj06MSJE7y2y2lZaWbtu2TVVWtBVCwWjxkzhiu
RyWSZmZngf8Yfh6+PYsWNohUkB2oOh+UkAwzBZWVk4Vb37duF93+F0+sNohUkAwzBZWVk4Vb37duXN54ol8v79esHRMo4w9bbtd
u1a9e4cLWOusKioyJGCsPv+JsA8wcjl8uTkZPCWZWVlsOm13XfmpSUwLJsVlYWTzBm
W3ZsiVQs9lsR48e5Y2yicXivn37NmzYEMyvn58+fBPamoqBCcwI9C03RS
UhLoYKDT6Y4fP+7Lsu6CsxAxk58xSUxMLC8vB0uAmG5AQEB0dHRhYSFQW7FiBVALDg7Oy8sDar7t/Oy
V1NRUsOiL6vyMkpCQoNPpwLpo8nNERER+fj5QQzs/e01+PnDgAIhpisXiAwcOADXMzs9emTNnjhuDDR
s2gIGBgYGnT58GagcPHuQtJPA00ORnTLwmP48ZM0bYbF7x2vkZkxkzZoDZvHZ+RlEqlcePHwdjvXZ+z
sjIEHxp5L+EBALBbyAOi0Ag+A3EYREIBL+BOCwCgeA3CE/7oGkaTUO1Wq0gIiASieRyRyRyObtP1p
BUsolUqcjXO3222xWEBSiM1mQwNYCrkcp6qn3W5Hk1oVCgUwxu12g0UtFotcLgdXIRKJgJrZbObP1p
HWJY1m81oogywTaFQWK1WoMYWDPog7HY7UHM6nSqVinfvoK8R1Eh7E0ltdeBBUFQkCsUuJ3TfB9
jY1dsWIFiMUcOXJk4MCBXElQUQUNDChQtB9efc3Fyg5nK5QFlun1NUVPTOO++AAD753797gIRqNxrlz54J
71b59e6Bms9nWr18P2ncnJCTs2bOH66EoisoaosrBlB+yV6Ojo7777DpxNATXO8bFarZ999hmHx+/evV
tYtiY40uFJh5wyZQrv2RSn04lmsL7MaDSatLQ0UKs+MzMTJPB7Pn/QwtZgMLwsnZ/Dn8CVWK1WtOOuV
qvt2rWrsJIgcU8QMNDlcl26dIRR72kSqXq0qULNfz888/g7NpUVFRrVq1atSoEVd4
/fp1oPY7YLVac3JPn15QUMAVJiQkALXq6uq5c+eCCdVqddeuXQX/CAIoFIrOnTu
Dcz2CYVn2ypUrQKhSqbp27Qrqxg18GZrPl5ua+kH7IzxWJRNKuXTsgLC4uBWVX/99dfnbQzZwyIQCH
4DcVgEAsFvIA6LQCD4DcRhEQgEvwFr093hcFy4cAG8yFYtJLwpVaq+ffqAFZqtWrVK165dUK1KlSpz5swBZeYYhtHpdEFBQbzhUTCvb9/eoA9uyKVq0aAH0+R7e3osZ26wbK+/v33n375twUEFBwVtb29YpsKu5c+eCOTvn59HWg7udnqQAF2KiIPj5IL6fHNM8A6B3iAWKDwlIx2OyD6bQk6nk1cvL88HBwaDsxQ39SSBGH/GjmY2+1h4HqBkC1x7PkWiiuFRCS4T5yVnP7vYWpKFWsvVL4EHbVYsYVyyQJqK4bPgfjRmQwURLoXMtrDzPrEFCzxP/mqYTU7
kn5CkVWMhK2rxctNG/9ehHtmWrtu9dfrN9BmIlL9ehNJs7yq8W4lDtZTfsIB9F8lWnAknyNEpEGKcEGzDxBfkxrHr0ruTJxwf2UN7fTfpgRtQ8n3Phf1UhlOMq
effVVEJzRaDRJSUlglRs3bqD7uIKJo4Gd4lhmAsXLjzXtuEKhaJ79+sVVVUCxoaGhoIHQdN027ZtQR
i6oqLi8uXL4FRE69atK1MlDF5Ig/oKVAoKCgoKSkJvP+3bt169OgRVyKKyzt27AgqDsTHxPbHA5HXl4eOK9w9erVZ
7xEH4GTcFhRUdGhQwZQfZ/2X06NEsy3LVLBYKIaN3/n56tWrUVFMkF89NFH1v/L48PgW8KCAgYMYMI
```

```
2WwGmi6XC1hy8uTJkJAQsMTGjRvBwIsXL8bExAC1hQsXAjWbzcYwDFgiLS0N2KZWq0+dOgXG7tq1S6l
UcucPDAzcv38/UDt+/DgahF27di1Qu3z5MlrBmaZpcAlSqRT9IzRp0iRwCQaDARTv90RdwWwqleqnn3
4CYxmGsSL89a9/BbM1bty4pKQE5/1EcblcYP6ysjLU4P79+9vtdmFLoDAMY7PZ0EvDYePGjeDWhYSEn
Dx5Eixx9OhRT1OP35DL5Xv37kUtQZeYPHkyuPzY2Ni7d+8CNYfDAWYrLy9v164dMK8ulbvrkvyMu6rD
4QDnxIT9GnoabrcbXQJ/LMgIl8lkaBUOmqblcjkqR2ezPwEdy5VIpVLUYIqihNWloShKKpWCsTRN2+1
27k8nlmXFYjFqCepiJBIJek9QNeYJAgz2Csuy4IYwDIMekvScJQZCwedLvUI/gStxOp28j76OePy1sL
GeZ82ViMVi9Mil5w5zNSmKwrzDqIuhKMrj8njNE/xt+hyyh0UgEPwG4rAIBILfQBwWgUDwG4TvnLndb
pZludsinj14r2q82Z6eqTATm70uAWbDUXuaJYLTNXGu1KMmbP6ngXnrBN9hwbPhB5HR2Z53eVh8PNu9
OJqYd/hpY2v559OWoCjK64uNY8nTvk3MzT50fq93ybfPUbjDOnv27FtvvcWVMAyDJhNdv379nXfe4d1
S1Wq1K1eu5N3/q6ys/PLLL0E3zYMHDxYVFXElTqfz3r17YOy5c+eGDx9e+/yeJQTvL27btu3ChQu8ao
WFhcLm90rTpk23bNkCSg60bdsWqMXEExGzYsIH30vR6/VdffaXT6WpXUyqViXYtAk2DUSiKQtPQvJKSk
jJgwACuRKVSoampL4SCgoJ58+bxnvaIiIhYtGgRyNbGpEePrt27eJKaJpG88Zbt269fft2rotxu91Z
WVmgWHtcXNy8efNAYX6U8vLy9957D3x0Q4cOHT9+PK/BXbt2/BXbt2/eSTT7gSm82Wmpp6/fp1rjAxMfGrr74
C2ZodO3bknf+p4IQSKyoqWrduLXwNDLzWdEcpKioCZfl/HzZt2gQsuXbtGmafZxw0Gk1ubi5YYYu/evc
DRSySSgwcPYoeAhVBSUoLWIUCPNfgdNTU1r732GruzGMNubm5aCkRlLi4uOLi4t/lav4XhmGGDh0KL
Gnbtm1lZSXQxGzj1nTfdSoUUDNZDL17NkTqPXq1ctisfjwesk9QNeYJAgz2Csuy4IYwDIMekvScJQZCwedLvUI/gStxOp28j76OePy1sL
```

```
s3zPyewLqjxUIvBoOaMUqlEDb5165ZOpwMWKhQKoIYP2cMiEAh+A3FYBALBbyAOi0Ag+A3EYREIBL9B
ePJzkyZNkpOTn2s/8aysLDR31IfQNP2Xv/wFZPA+evRo165dIGA3cODAFi1agLEbNmwAsdhu3bp16dK
FK1GpVP/1X/8FzgTI5fKlS5cKM/jNN9+Mi4vjCouKinbu3AkCMWgL7uDg4NGjR4PNeGCtJ76+bds2UE
zy3LlzOObFxcV9+umn3FiESCQC1j6Nvn37gpLWer1+69atgktxPm+OHj0KkskZhhk7diwI1+bk5IBEX
680b958xowZXIlUKgWZ/17xNNy9c+cOV1hcXIyzE69WqydPngx22YUlSHvqiG7fvh18sA6HA23gmpCQ
IGyJgLp0fn7rrbdYln2WJMpnBjxCT9OEwsJC3oGVlZVor3YUmUx2+PBhMPbEiRNoiA2z83NqairOdaW
lpaUF9LoAAAxFSURBVAl6Vt47P2MmPyckJOh0Ol7bXlTy87p164Bafn4+Gq/EBE1+9tr5GYWm6b1794
Kxden8nJqaCtQwk5+9cvjwYVBHBB/fJj9jQjo/EwiE/1yIwyIQCH4DcVgEAsFvIA6LQCD4Db4skezZ2
hTWdIymaY1Gg5kACbBarSDRV6/XC277KhaLQ0JCQCdhwTudDMNUV1eDJB7eHsIepFIpSK8Vi8VWqxVk
yRoMBrTHZ2BgILA5KChIr9fz5o7o9Xq1Wg1ys202G7jDLMsaDAZgCSYGgwHNakKzf/V6PaqmUChwEn3
RKARFURqNBlyX0+nEzERBUalU4CVRq9UGgwHkhKEp6CzL6vV6cLEymQw8a69IJJLQ0FCQ7WQ0GnHy/N
E7jEllZSX6NaEvp+fzf67tln3ssOx2+6effooZAgfEx8enp6cLa1e5e/fuv//971wJwzD37t0TMJWnW
uvBgwfBpwLSrfGprKycMGEC6KRfWVmJM/a1115btmwZ18W4XK5ly5bNmzePq2Y2m9FO4jNmzADNdR8/
fjxp0iTe71Ot4ThUN+1a9fXX3/NlVgslqlTpworEmB5AhB+++23Gzdu5Eq8towdwdO3bshx9+yLs
E+iIpFIpVq1YB93H69Olp06YJ+8amT58+YsQIrqSsrCwlJQV0ZkVzoQ0Gw8SJE4GzGzRo0OLFi3kXbd
++/eHDh7kvp9vt/vjjj3/55Rfesd9999327dt51VBcLhf6NfXq1Qt8dFarcqUKbzp5XXEx7+wCgsL0
RNAmAiry+H5/gUviqJSqVq2bOmr2Vwu1+3bt4V5T7T7VanZiYCDo/Y15sgwYNWrVqxZVIpdI7d+7w1hK
iIh45ZVX4uPjucLc3FygxrIs6Is6fwKsWEREBrgsTmqYbNWqELIr4IGFMTAyw5O7duwUFBByZ8wz
DgGNTnj+TOIsGBQWBl5l5NlWcw6BByVPwNHEQavVgss9qEVMMGw8BMzcrV64lOGGiAzurjDLs
k2aNNFqtVy18vbgeWWK1WNNKbK+gdRnG73fKNZGCjXYRGW3zGtxmqLe8Xe+6WXS2MUgv
+IFMTAyw5O7duwUFBbyZ8wz
DgGNTnj+TOIsGBQWBl5NlWcw6ByVPwNHEQavVgss3m82CewvgQ/awCASC30AcFoFA8BuIwyIQCH4DcV
gEAsFveKk7PwtGIpEkJSXx9ogWi8UgHFZHrly5sm3bNq5Er9fjfjGKQSCQ9e/YEQfeIiIMjAzurjDLs
k2aNNFqtVy18vLyEydO+CqcbLPZ/v3v3vf4PbgkZ+xWJxz549QYVoFJZlT506BfZ6Q0NDe/bsydu0
ZmZmghD+tWvXBuC6WlpW+99RY4FHLv3j30IQ4ZMgTcYYPBANRcrZnZ/fB7OJxeIePXoIq0M
bHByclJQkuOguwG63Hz9+kBmlsicY9BDBbDDSOSOVU1xvlTVFVUcPWAB33q0Zh9Dk
4FXDCTn9HOz8HBwXl5eXVOtKwNr8nPgtFoNLm5uwCJvXv3goenNn/y8YcMGoHbz5k3BVZhRMJOfHQ7Ho
EGDwNjOnTt7zo7VTl2SnzHp37/3W7nLup0OpOTk4Fahw4dTc9MC8lxiGAedX
PH3kKysrecdigpn8rFQqjx8/7qtFnwb5LyGBQPAbiMMiEAh+A3FYBALBbyAOi0Ag+A1/zCihV1wulw9
75NntdrlcDpLpHA4HHGq2Ty+Ugqdv5BN4laJpWKpcjCWxWIyzHaWIyZH26320HenM1mwzTYqyUglVqhUDgcDr
CER01YsovdbgeWWK1WNKnbK+gdRnG73Xa7Hc2mRpHJZOYBn8g/PAL2FhYWlpaaCYcnp6+tatW7mSkJCQ
lStXgkan+/fvX758ufG8e8SnTt33r9/PvvP/P+j8jlmsdvny5T5PNCXxhoaunDfNN/ezDtbv379Pvv
sM67EEztXX78epPJ5wXyoAUCaXZaOCh2kYgd5OycXyySaDXYKhQKSTuPbAqcqwY
cO4krKysilTpoBYaGwgqBJgnrnWMGuyEqeGcUWNcEYTOpDSpUQYOvaOQsMmYOiCuYDlWiw1JobUPq6Ooo
c8eBq6tnhmWCQkJRZOsX+trH4WwOm6ShA5Q8cOrWq+DDA0aNH
gZpMJuvcuTNTbHIu7116xbOEmFhYT179+vg3I9Lm8OsAyoRojc2H9Jqgbj1k8Z/DX+XbTvbeGaQBdHDkmVz4KuLqLhLjwc
rVKlUgm/v7du3s7KyBAyUSqWdOnQVqWpGoxEnSVgkEj3vr04qlXbs2BH0NEExmUyaUypqak4DSy8jj
1z5gwQvvnmm0BaRNgVvijIHhaBQPAbiMMiEAh+A3FYBALBbyAOi0Ag+A2+3HQXi
URt2rQtQRNjYuLo43G/Zp1K9fH+QwBj1B2GxeAd68OcjqbNiwoWMGQJwCDY2JigMGBgYFcFob9Hy8vJR
RF
tWrVCqQcazSHj16gLTbmzdvgvQ4YGBg9x+g27Kzy3NJdt6gvKz3qavTdGkHhD3i3wZ8zt7z16Hh+K
UG8Xn8ePHN2/e5EosFgsoxBGsoXlxHEhIScDJn79+/j1NvtqSk5MaNG1yJr
PB0KFDB9C/WiKRADVmzZoJh
ttvcyV1MXjgwIF9+vQBQnS2nJyc0aNHc89hSSSHTt29OvXj6vWrFmzHTt2gLHvvffeDz/8wJXY
HTsWqKFHeLRa7dq1a0HHzZ98m3N+6dWvUqFFAKKx3ydPIzMx8//33YyCnUwM+H3zwD6/
88ssv7777LoX8SR/PvHxYt+SYYMeWvKYJFHzGOHRI7pEgvCfcKPQO//Oo8IlALBwxXk+9CuVfhvJ6F
2+4bogfsJzXUIqlYIfHXUB02DPuT6uw2IYBv17qTJbvItSF2UUoCWXhdnz4ahs6i7vvc3TMaHzSiP1Lo9
xKYxVswf4YLNtjz0eE8L/S2OxwO9FlTFCXMEpFIJOzHyrOt8rwXIBAIBF9BHBaBQPAbiMMiEAh+A3FYB
ALBbyAOi0Ag+A1/zCihV1wulw9DzsUkEZrn3/kOiOYjsfUM3NFYv3A1kyHo+ntqU8/aA
```

```
ULFoDWpMXFxWAXk2GYRYsWgQ7JXunQoUNGRgav2o8//rhr1y4BBiuVytTUVJA3GxUVBeKJNE3PmjVr/
PjxXGF+fn5qaio4h/H+++/37t2bK9HpdHPmzAGlxFFqamqmT58ODoXcv3//2a/pqRQUFCxcuBB0q540
aRII15aWls6ZMwccngCnK55G9+7dcZ4XitvtPnHiBE6d+0uXLgFJRUVFSkoK2HQHKaieMzfz5s3jTS9
3uVw7d+78/vvveS355JNPunTpwqvmHZw6yl5rur8QvNZ0R6msrGzbti3vbDKZ7PDhw2DsiRMnfJiaHx
UVVVBQwGuwXq/3bX4pWtPdK3PmzAEDIyIi8vPzccYKJjs7Gz10tm7dOqD2Qmq6eyU3NxdNnE5LSwNqB
QUFUVFRvIuGh4dfu3bNVzfTa0133xIaGnrlyhVeS0wmE2aufkZGhuDrJXtYBALBbyAOi0Ag+A3EYREI
BL+BOCwCgeA3YEUJaZpOSEh4/sbwU69ePZyUCJqmmzRpwk1q8YpUKkVzpFUqVWJiouAWvoDw8HBMg+P
j432Y7IaZgxoZGQnCKaGhob7qGPw0lEplYmIiKLmLGiyVStFOwr4lLi4OJ8ztMRgkCYM23b8ZzNutNj
g42LdF4v/0pz8915iYVqvFMVgkEjVq1AgnYRun9OvToHDOTbAsazQafdUMvS6IRCK1Ws2bi4tpMEVRg
YGB4Pt0Op1Go9FXx2d8azA+gYGBOHmdFosFuGZMg+uC1zusUqnAV8EwTE1NzXNNT5NIJEFBQbw+y+Vy
1dTU+MpgkUgUFBTkq+xXT3jOt3ndAIqi1Go1r8Fut9toNOIksWO+nN6NIacQCQSCv0D2sAgEgt9AHBa
BQPAbiMMiEAh+A3FYBALBbyAOi0Ag+A3EYREIBL+BOCwCgeA3EIdFIBD8BuKwCASC30AcFoFA8BuIwy
IQCH7D/wsAAP//41hLy/71ZdkAAAAASUVORK5CYII=",
        "order": {
            "merchantId": "305142568",
            "clientId": "",
            "paymentType": "E_COMMERCE",
            "merchantTradeNo": "16561111111000030002",
            "payId": "01JXHW8QAEZXSA5HTJYT3AS357",
            "status": "INIT",
            "amount": "0.2",
            "currency": "USDT",
            "currencyType": "crypto",
            "createTime": 1749725503,
            "paymentTime": 0,
            "finishTime": 0,
            "refundOrders": []
        },
        "customer":{
            "uid":"123456",
            "externalUserId": "test@gmail.com",
            "userName": "john li",
            "registerTime": "1739178959",
            "kycCountry": "AUS"
        }
    }
}
```

6.1.2 Payment Result

- **Data Type:** JSON (content-type: application/json)
- **Request Method:** GET
- **Path:** `/v5/bybitpay/pay_result`

Query Parameters:

Field Name	Type	Required	Description
merchantId	string	Yes	Merchant ID
clientId	string	No	Merchant Client ID
paymentType	<Payment Type>	Yes	Payment Type
merchantTradeNo	string	No	Merchant Order Number if paymentType is `E_COMMERCE` ,Merchant Refund Order Number if paymentType is `E_COMMERCE_REFUND` (required if payId is not provided)
payId	string	No	Bybit Pay ID if paymentType is `E_COMMERCE` ,Bybit refund ID if paymentType is `E_COMMERCE_REFUND` , (required if merchantTradeNo is not provided) **【Recommend】**

Request demo

- Query Pay Order

```
{{base_url}}/v5/bybitpay/pay_result?merchantId=305142568&paymentType=E_COMMERCE&merchantTradeNo=fe43f6b0-fb6c-4f17-af07-d069746ec760
```

- Query Refund Order

```
{{url}}/v5/bybitpay/pay_result?merchantId=305142568&paymentType=E_COMMERCE_REFUND&merchantTradeNo=b4b7c249-3ff1-4a85-9fc9-b95420a3cec0
```

Response:

Field Name	Type	Required	Description
retCode	number	Yes	Return Code, 100000: success, not 0: failed, See<Error Code>
retMsg	string	Yes	Return Message
result.order	<PayOrder Type>	Yes	Bybit PayOrder Detail
result.customer	<Customer Type>	Yes	Bybit user info

Response demo

- Pay Order Response

代码块

```
{
    "retCode": 100000,
    "retMsg": "success",
    "result": {
        "order": {
            "merchantId": "305142568",
            "clientId": "",
            "paymentType": "E_COMMERCE",
            "merchantTradeNo": "fe43f6b0-fb6c-4f17-af07-d069746ec760",
            "payId": "01JN5WJNK8ZYK8B7JGKTVNXCPN",
            "status": "TIMEOUT",
            "amount": "100",
            "currency": "USDT",
            "currencyType": "crypto",
            "createTime": 1740733241,
            "paymentTime": 0,
            "finishTime": 0,
            "refundOrders": []
        },
        "customer":{
            "uid":"123456",
            "externalUserId": "test@gmail.com",
            "userName": "john li",
            "registerTime": "1739178959",
            "kycCountry": "AUS"
        }
    }
}
```

- Refund Order Response

代码块

```
{
    "retCode": 100000,
    "retMsg": "success",
    "result": {
        "order": {
            "merchantId": "305142568",
            "clientId": "",
            "paymentType": "E_COMMERCE",
            "merchantTradeNo": "fe43f6b0-fb6c-4f17-af07-d069746ec760",
            "payId": "01JN5WJNK8ZYK8B7JGKTVNXCPN",
            "status": "PAY_SUCCESS",
            "amount": "100",
            "currency": "USDT",
            "currencyType": "crypto",
            "createTime": 1740733241,
            "paymentTime": 1740733241,
            "finishTime": 0,
            "customer":{
                "uid":"123456",
                "externalUserId": "test@gmail.com",
                "userName": "john li",
                "registerTime": "1739178959",
                "kycCountry": "AUS"
            },
            "refundOrders": [
                {
                    "refundId": "D41717457E604297B96E1E7E8D238302",
                    "refundType": "MERCHNT_SELF_REFUND",
                    "merchantTradeNo": "fe43f6b0-fb6c-4f17-af07-d069746ec760",
                    "merchantRefundNo": "b4b7c249-3ff1-4a85-9fc9-b95420a3cec0",
                    "payId": "01JN5WJNK8ZYK8B7JGKTVNXCPN",
                    "refundStatus": "REFUND_SUCCESS",
                    "amount": "100",
                    "createTime": 1740748919
                }
            ]
        }
    }
}
```

6.1.3 Payment Notify

- **webhook_url:** Provided in `create_payorder` API
- **Webhook Body Type:** `PayOrder/RefundOrder/Payout Type`
- **Webhook Method:** `POST`
- **Authentication:**
    - Share the IP whitelist with each other
    - Bybit Pay will encrypt the callbacks via RSA_SHA256. Partners can provide the public key, sign date put in result head. For details, please see <Signature Algorithm>
- **Headers**

Content-Type: `application/json`

timestamp: xxx

publicKey: xxx

signature: xxx

- **RequestBody**

Field Name	Type	Required	Description
paymentType	<PaymentType>	Yes	Payment type
-	<PayOrder>	No	Returned if paymentType is `E_COMMERCE`
-	<RefundOrder>	No	Returned if paymentType is `E_COMMERCE_REFUND`

Response demo

- Notify PayOrder

代码块

```
{
    "paymentType": "E_COMMERCE",
    "merchantId": "12",
    "clientId": "xx",
    "merchantTradeNo": "123456",
    "payId": "123456",
    "status": "INIT",
    "amount": "100",
    "currency": "USDT",
    "currencyType": "crypto",
    "createTime": 1740488831,
    "paymentTime": 1740488831,
    "finishTime": 1740488831,
    "customer":{
        "uid":"123456",
        "externalUserId": "test@gmail.com",
        "userName": "john li",
        "registerTime": "1739178959",
        "kycCountry": "AUS"
    }
}
```

- Notify Refund Order

代码块

```
{
    "paymentType": "E_COMMERCE_REFUND",
    "refundId": "123",
    "refundType": "MERCHNT_SELF_REFUND",
    "merchantTradeNo": "123456",
    "merchantRefundNo": "123456",
    "payId": "123456",
    "refundStatus": "REFUND_SUCCESS",
    "amount": "1739178959",
    "createTime": 1740488831,
    "customer":{
        "uid":"123456",
        "externalUserId": "test@gmail.com",
        "userName": "john li",
        "registerTime": "1739178959",
        "kycCountry": "AUS"
    }
}
```

5.1.4 Payment status mock(only sanbox testnet)

- **Data Type:** JSON (content-type: application/JSON)
- **Request Method:** POST
- **Path:** `/v5/bybitpay/paystatus/mock`

Request Body:

Field Name	Type	Required	Description
merchantId	string	Yes	Merchant ID
clientId	string	No	Merchant Client ID
paymentType	<Payment Type>	Yes	Payment Type
merchantTradeNo	string	No	Merchant Order Number if paymentType is `E_COMMERCE` ,Merchant Refund Order Number if paymentType is `E_COMMERCE_REFUND` (required if payId is not provided)
payId	string	No	

			Bybit Pay ID if paymentType is `E_COMMERCE` ,Bybit refund ID if paymentType is `E_COMMERCE_REFUND` , (required if merchantTradeNo is not provided)**【Recommend】**
status	string	Yes	Order Status

Request demo

代码块

```
//E_COMMERCE
{
    "paymentType": "E_COMMERCE",
    "merchantId": "12",
    "clientId": "xx",
    "merchantTradeNo": "123456",
    "payId": "123456",
    "status": "PAY_SUCCESS"
}

//E_COMMERCE
{
    "paymentType": "E_COMMERCE_REFUND",
    "merchantId": "12",
    "clientId": "xx",
    "merchantTradeNo": "123456",
    "payId": "123456",
    "status": "REFUND_SUCCESS"
}
```

Response:

Field Name	Type	Required	Description
retCode	number	Yes	Return Code, 100000: success, not 0: failed, See<Error Code>
retMsg	string	Yes	Return Message
result.order	<PayOrder Type>	Yes	Bybit PayOrder Detail

Response demo

- Pay Order Response

代码块

```
{
    "retCode": 100000,
    "retMsg": "success",
    "result": {
        "order": {
            "merchantId": "305142568",
            "clientId": "",
            "paymentType": "E_COMMERCE",
            "merchantTradeNo": "fe43f6b0-fb6c-4f17-af07-d069746ec760",
            "payId": "01JN5WJNK8ZYK8B7JGKTVNXCPN",
            "status": "PAY_SUCCESS",
            "amount": "100",
            "currency": "USDT",
            "currencyType": "crypto",
            "createTime": 1740733241,
            "paymentTime": 0,
            "finishTime": 0,
            "refundOrders": []
        }
    }
}
```

5.1.5 Payment Fx Convert

- **Data Type:** JSON (content-type: application/JSON)
- **Request Method:** POST
  - **Path:** `/v5/bybitpay/fx/convert`

Request Body:

Field Name	Type	Required	Description
merchantId	string	Yes	Onboarding merchant UID
merchantName	string	No	If not filled in, we will show users the name of your company doing KYB on the Bybit platform
clientId	string	No	Merchant Client ID
paymentType	string	Yes	Payment Type

orderAmount	string	Yes	Order Amount
orderCurrency	string	Yes	Order Currency (uppercase, e.g., EUR, USD, USDT)
orderCurrencyType	string	Yes	Currency Type (`fiat` or `crypto`)
settleCurrency	string	Yes	settleCurrency ,USDT/USDC/BTC
settleCurrencyType	string	Yes	Currency Type (`fiat` or `crypto`)

Request demo

代码块

```
{
    "merchantId": "305142568",
    "merchantName": "uquid",
    "paymentType": "E_COMMERCE",
    "orderAmount": "0.001",
    "orderCurrency": "ETH",
    "orderCurrencyType": "crypto",
    "settleCurrency": "USDT",
    "settleCurrencyType": "crypto",
}
or
{
    "merchantId": "305142568",
    "merchantName": "uquid",
    "paymentType": "E_COMMERCE",
    "orderAmount": "78.01",
    "orderCurrency": "EUR",
    "orderCurrencyType": "fiat",
    "settleCurrency": "USDT",
    "settleCurrencyType": "crypto",
}
```

Response:

Field Name	Type	Required	Description
retCode	number	Yes	Return Code, 100000: success, not 0: failed, See<Error Code>
retMsg	string	Yes	Return Message

result.	object	Yes	Return Result information
result.quotationId	string	Yes	Return Result quotationId to create order
result.settleCurrency	string	Yes	settleCurrency ,USDT/USDC/BTC
result.settleAmount	string	Yes	Order settle Amount

Response demo

代码块

```

{
    "retCode": 100000,
    "retMsg": "success",
    "result": {
        "quotationId": "1230000001"
        "settleCurrency": "USDT",
        "settleAmount": "10.00"
    }
}
```

6.2 Order Refund

- **Data Type:** JSON (content-type: application/json)
- **Request Method:** POST
- **Path:** `/v5/bybitpay/refund`

Request Body:

Field Name	Type	Required	Description
merchantId	string	Yes	Merchant ID
clientId	string	No	Merchant Client ID
list	array <RefundOrderItem>	Yes	Refund Order Request Item

Request demo

代码块

```
{
    "list": [
        {
            "refundType": "MERCHNT_SELF_REFUND",
            "merchantTradeNo": "fe43f6b0-fb6c-4f17-af07-d069746ec760",
            "payId": "",
            "merchantRefundNo": "b4b7c249-3ff1-4a85-9fc9-b95420a3cec0",
            "refund_amount": "101",
            "env": {
                "browserVersion": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36",
                "device": "xxde",
                "ip": "127.0.0.1",
                "terminalType": "APP"
            }
        }
    ],
    "merchantId": "305142568"
}
```

Response:

Field Name	Type	Required	Description
retCode	number	Yes	Return Code, 100000: success, not 0: failed, See<Error Code>
retMsg	string	Yes	Return Message
result.refundOrders	array <RefundOrder>	Yes	Bybit refund order detail

Response demo

代码块

```
{
    "retCode": 100000,
    "retMsg": "success",
    "result": {
        "refundOrders":[
        {
            "refundId":"123",
            "refundType":"MERCHNT_SELF_REFUND",
            "merchantTradeNo":"123456",
            "merchantRefundNo":"123456",
            "payId":"123456",
            "refundStatus":"REFUND_SUCCESS",
            "amount":"100",
            "amount":"1739178959"
        }
        ]
    }
}
```

6.3 Payout

- **Data Type:** JSON (content-type: application/json)
- **Request Method:** POST
- **Path:** `/v5/bybitpay/payout`

Request Body:

Field Name	Type	Required	Description
merchantId	string	Yes	Onboarding merchant UID
merchantName	string	No	If not filled in, we will show users the name of your company doing KYB on the Bybit platform
clientId	string	No	Merchant Client ID
paymentType	string	Yes	Payment Type
merchantTradeNo	string	Yes	Merchant Order Number
amount	string	Yes	Order Amount
currency	string	Yes	Order Currency (uppercase, e.g., EUR, USD, USDT)
currencyType	string	Yes	Currency Type (`fiat` or `crypto`)
combined	bool	No	Whether combined payment is supported (`true` or `false`), default `false`。 eg: When merchant's USDT assets are insufficient for the order amount, can BTC or

			other crypto currency assets be exchanged for USDT
webhookUrl	string	No	API URL to call when order is successful or failed (max 256 characters)
payee	<Payee>	Yes	Bybit payee
env	`Env Type`	Yes	Transaction Source
riskInfo	<RiskInfo>	No	Risk info for order pay
mccCode	string	Yes	Merchant category codes,ISO 18245:2023

Request demo

代码块

```
{
    "currency": "USDT",
    "currencyType": "crypto",
    "paymentType":"MERCHANT_PAYOUT",
    "merchantId": "1546593",
    "merchantName":"merchantName",
    "merchantTradeNo": "123456",
    "amount": "0.00001",
    "combined":true,
    "payee":{
       "uid":"104326306"
    },
    "webhookUrl": "https://qatools.hrmocks.com/callback_catcher/save/bybit",
    "env": {
        "browserVersion": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36",
        "device": "400017492767368721919",
        "ip": "187.12.0.1",
        "terminalType": "WEB"
    }
}
```

Response:

Field Name	Type	Required	Description

retCode	number	Yes	Return Code, 100000: success, not 0: failed, See<Error Code>
retMsg	string	Yes	Return Message
result	object	Yes	Return Result information
result.order	`PayOrder`	Yes	Bybit PayOrder Detail

Response demo

代码块

```
{
    "retCode": 100000,
    "retMsg": "success",
    "result": {
        "order": {
            "merchantId": "1546593",
            "clientId": "",
            "paymentType": "MERCHANT_PAYOUT",
            "merchantTradeNo": "123456",
            "payId": "01JXMVAX2K7YNP1081ST1KSCCC",
            "status": "INIT",
            "amount": "0.00001",
            "currency": "USDT",
            "currencyType": "crypto",
            "createTime": 1749825188,
            "paymentTime": 0,
            "finishTime": 0,
            "refundOrders": []
        }
    }
}
```

~~6.4 Settlement report~~

- **Data Type:** JSON (content-type: application/json)
- **Request Method:** POST
- **Path:** `/v5/bybitpay/settlement_record`

Request Body:

Field Name	Type	Required	Description
merchantId	string	Yes	Merchant ID
clientId	string	Yes	Merchant Client ID
PaymentType	string	Yes	Payment Type
settleAccountBeginDate	string	Yes	settle account date , like 2025-09-10
settleAccountEndDate	string	Yes	settle account date , like 2025-09-12
start	int	Yes	start ,default 0
limit	int	Yes	limit ,default 100

Request demo

代码块

```
{
    "merchantId": "123456",
    "clientId": "123456",
    "paymentType": "E_COMMERCE",
    "settleAccountBeginDate": "2025-01-01",
    "settleAccountEndDate": "2025-02-01",
    "start":"0",
    "limit":"1000"
}
```

Response:

Field Name	Type	Required	Description
retCode	number	Yes	Return Code, 0: success, not 0: failed
retMsg	string	Yes	Return Message
result.list	Array< `SettlementOrder Type` >	Yes	Settlement Order List

Response demo

代码块

```
{
   "retCode": "100000",
    "retMsg": "success",
    "result": [{
        "paymentType": "E_COMMERCE",
        "settlementTaskNo": "1111111",
        "orderBusinessDate": "2022-10-10",
        "settleAccountDate": "2022-10-10",
        "settlementCurrencyCode":"USDT",
        "totalPayAmount":"700.00",
        "totalOrderAmount":"1000.00",
        "totalRefundAmount":"100.00",
        "totalFeeAmount":"100.00",
        "settlementDetails":[{
            "paymentType": "E_COMMERCE",
            "orderAmount": "100.00",
            "OrderCurrency": "USDT",
            "orderType": "order",
            "settlementCurrencyType": "USDT",
            "merchantOrderNumber": "1739178959"
        },{
            "paymentType": "E_COMMERCE",
            "orderAmount": "200.00",
            "OrderCurrency": "USDT",
            "orderType": "order",
            "settlementCurrencyType": "USDT",
            "merchantOrderNumber": "1739178953"
        },{
        "paymentType": "E_COMMERCE",
        "settlementTaskNo": "1111111",
        "orderBusinessDate": "2022-10-11",
        "settleAccountDate": "2022-10-11",
        "settlementCurrencyCode":"USDT",
        "totalPayAmount":"800.00",
        "totalOrderAmount":"1000.00",
        "totalRefundAmount":"0.00",
        "totalFeeAmount":"100.00",
        "settlementDetails":[{
            "paymentType": "E_COMMERCE",
            "orderAmount": "100.00",
            "OrderCurrency": "USDT",
            "orderType": "order",
            "settlementCurrencyType": "USDT",
            "merchantOrderNumber": "1739178959"
        },{
            "paymentType": "E_COMMERCE",
            "orderAmount": "200.00",
            "OrderCurrency": "USDT",
            "orderType": "order",
            "settlementCurrencyType": "USDT",
            "merchantOrderNumber": "1739178953"
        }]
    }]
}
```

7. Data Models

7.1 Env

Field Name	Type	Required	Description
terminalType	string	Yes	Transaction source (APP, WEB, WAP, MINIAPP, OTHERS)
device	string	Yes	App or pc system device uquid user info
browserVersion	string	Yes	App or pc system browser version
ip	string	Yes	Ip infro

7.2 Good

Field Name	Type	Required	Description
shoppingName	string	Yes	shoppingName on bybitpay Partner,like Apple
mccCode	string	Yes	Merchant category codes,ISO 18245:2023
goodsName	string	Yes	Product name (max 160 characters)
goodsDetail	string	No	Product description (max 256 characters)

7.3 PayOrder

Field Name	Type	Required	Description
merchantId	string	Yes	Merchant ID
clientId	string	No	Merchant Client ID，support different Business type
paymentType	string	No	Payment Type
merchantTradeNo	string	Yes	Merchant Order Number
payId	string	Yes	Bybit Pay ID
status	string	Yes	Order Status， State Machine
amount	string	Yes	Order Amount
currency	string	Yes	Order Currency Crypto scale: 8 examples: 9.00000001USDT
currencyType	string	Yes	Order Type (`fiat` or `crypto`)
createTime	number	Yes	Order Creation Time (second timestamp)
paymentTime	number	Yes	Order Payment Time (second timestamp)
finishTime	number	Yes	Order Completion Time (second timestamp)
refundOrders	array<RefundOrder>	Yes	Refund Orders
remark	string	No	Payment remark ,max 50 characters

7.4 RefundOrder

Field Name	Type	Required	Description
refundId	string	No	Bybit refund ID (required if `merchantRefundNo` is not provided)
refundType	string	Yes	refund type
merchantTradeNo	string	Yes	Merchant Order Number
merchantRefundNo	string	No	Merchant Refund Number (required if `refundId` is not provided)

payId	string	Yes	Bybit Pay ID
refundStatus	string	Yes	Order refund Status
refundCurrency	string	No	refund currency,default order currency
amount	string	Yes	Order Amount
createTime	number	Yes	Order Creation Time (second timestamp)

7.5 RefundOrderItem

Field Name	Type	Required	Description
refundType	string	Yes	refund type
merchantTradeNo	string	No	Merchant Order Number(required if `payId` is not provided)
payId	string	No	Bybit Pay ID(required if `merchantTradeNo` is not provided)
merchantRefundNo	string	Yes	Merchant Refund Number
refundAmount	string	Yes	Refund amount, support partial refund
env	<Env>	Yes	
riskInfo	<RiskInfo>	No	Risk info for order pay
customer	<CustomerType>	No	Customer info

7.6 SettlementOrder

Field Name	Type	Required	Description
paymentType	string	Yes	PaymentType
settlementTaskNo	string	Yes	Merchant SettlementTaskNo
orderBusinessDate	string	Yes	order date , like 2025-09-09
settleCreateDate	string	Yes	settle create date , like 2025-09-10

settleAccountDate	string	Yes	settle account date , like 2025-09-11
settleAccountAmount	string	Yes	settleAccountAmount
totalPayableAmount	string	Yes	merchants order totalPayable amount
totalPayAmount	string	Yes	bybit pay to merchants amount
totalPrincipalAmount	string	Yes	totalPrincipalAmount
totalOrderAmount	string	Yes	merchants order amount
totalRefundAmount	string	Yes	merchants order refund amount
totalFeeAmount	string	Yes	merchants fee amount
previousRollingReseverAmount	string	Yes	previous Rolling Resever Amount
currentRollingReserveDeductionAmount	string	Yes	Current Rolling Reserve Deduction Amount
fixedReserveDeductionAmount	string	Yes	fixed Reserve Deduction Amount
settlementDetails	List<SettlementOrderDetail>	Yes	Settlement Order Detail

SettlementOrderDetail

Field Name	Type	Required	Description
settlementVoucherNumber	string	Yes	settlement Voucher Number
merchantOrderNumber	string	Yes	merchant Order Number
orderType	string	Yes	orderType, order/refund
originalOrderNumberRefund	string	Yes	original OrderNumber of Refund
orderStartTime	string	Yes	order Start Time
merchantNumber	string	Yes	merchant Number
clearingBatchNumber	string	Yes	clearingBatchNumber
orderAmount	string	Yes	total Order Amount
OrderCurrency	string	Yes	OrderCurrency

orderClearingCurrencyType	string	Yes	order Clearing Currency Type:CRYPTO/FIAT
OrderClearingExchangeRate	string	Yes	OrderClearingExchangeRate
principalClearingAmount	string	Yes	principalClearingAmount
principalClearingCurrencyType	string	Yes	principalClearingCurrencyType
principalClearingCurrency	string	Yes	Principal Clearing Currency
principalClearingExchangeRate	string	Yes	principalClearingExchangeRate
marginClearingAmount	string	Yes	marginClearingAmount
margin Clearing Currency Type	string	Yes	Margin Clearing Currency Type
margin Clearing Currency	string	Yes	Margin Clearing Currency
margin Clearing Exchange Rate	string	Yes	Margin Clearing Exchange Rate
feeClearingAmount	string	Yes	Fee Clearing Amount
feeClearingCurrency Type	string	Yes	Fee Clearing Currency Type
feeClearingCurrency	string	Yes	Fee Clearing Currency
feeClearingExchangeRate	string	Yes	Fee Clearing Exchange Rate
marginReturnClearingAmount	string	Yes	Margin Return Clearing Amount
marginReturnClearingCurrencyType	string	Yes	Margin Return Clearing Currency Type
marginReturnClearingCurrency	string	Yes	Margin Return Clearing Currency
marginReturnClearingExchangeRate	string	Yes	Margin Return Clearing Exchange Rate

7.7 PaymentType

Field Name	Type	Description
E_COMMERCE	string	Bybit QR Pay for e-commerce
E_COMMERCE_REFUND	string	BYBIT QR Pay refund for e-commerce
QR_PAY	string	Bybit QR Pay for stand qrcode pay
QR_PAY_REFUND	string	Bybit QR Pay refund for stand qrcode pay

MERCHANT_PAYOUT	string	Bybit pay payout for merchant

7.8 PaymentMethedDetail

Field Name	Type	Description
PaymentType	List string	Payment Type
fiatCurrency	List string	Fiat Currency
cryptoCurrency	List string	Crypto Currency

7.9 Customer

Field Name	Type	Description	
uid	string	bybit user unique id	
externalUserId	string	External merchant user ID	
userName	string	User name in Merchant	
registerTime	int64	User register time in UTC timestamp (second precision)	
kycCountry	string	3 ISO country codes, like AUS	
remarks	string	External order remarks	

7.10 RiskInfo

Field Name	Type	Required	Description
terminalType	string	Yes	Transaction source (APP, WEB, WAP, MINIAPP, OTHERS)

7.11 RefundType

Type	Description

MERCHNT_SELF_REFUND	Merchant refund by Kyb fundingaccount
STAND_SETTLEMENT_REFUND	Settlement refund on T1、T7 report

6.12 Payee

Field Name	Type	Required	Description
uid	string	Yes	Payee bybit uid

8. MccCodes

Table A1 Codes defined by ISO 18245

9. Signature Algorithm

8.1. Encrypt

1. Get the current timestamp `${timestamp}`
2. Combine `${timestamp}` and request body (if the http method is `POST`, the request body is `json` text, if the http method is `GET`, the request body is unescape `query` string, like `name=foo&age=18` is valid, `name%3dfoo%26age%3d18` is invalid) into a new string `$content`
3. Sign `$content` with `sha256` and `PKCS1V15, 1024`
4. Encode the signed bytes content to `base64`

e.g. We sign the following data. The timestamp is `1751441054`, the RSA key is

代码块

```
-----The following key pairs are for testing only-----
-----BEGIN RSA PUBLIC KEY-----
```

```
MIGJAoGBAOFSnhqtu40TOtok+yXeB+O76PXb/VAJU4Yih6hViOdSGd7imWmCSZyP
psl3TmLhUoB+rTzYDdYrlYYng6cVn6yUhxjpMWD4Qp5K4GzjvUM0f+AxlKYMj8OQ
AgDPmZG1a5ydFrje4PLytC7sUw3GP4TTk8xg6iMHmYPdRDv7AEWdAgMBAAE=
-----END RSA PUBLIC KEY-----

-----BEGIN RSA PRIVATE KEY-----
MIICXwIBAAKBgQDhUp4arbuNEzraJPsl3gfju+j12/1QCVOGIoeoVYjnUhne4plp
gkmcj6bJd05i4VKAfq082A3WK5WGJ4OnFZ+slIcY6TFg+EKeSuBs471DNH/gMZSm
DI/DkAIAz5mRtWucnRa43uDy8rQu7FMNxj+E05PMYOojB5mD3UQ7+wBFnQIDAQAB
AoGBAIqpeCi83516xw32EiSEleBdBIlA34vAoYj7ePZr5PtN+CPE2Von1rslYCO9
axzIwyfWTTATWbiCS9sqOAe78+1t3shU0ryXelWQ95LOdiMp0ZvU3ya6MzSVuiuL
6gOQYu8kzRqCzqvyMNdAHFMmP0iESefYegaiU46yQo2ZZAtVAkEA/N6gl4tYJBef
e/Gxi7qhuIKz0mvfA/yieRuiVhFb25oJaagU6hBuzegmbs45hh+n4bw8AA9k0bvi
3IBIdV/CbwJBAOQcsOPf+lrs4PEJYUQ8/eqiG7tgIUeXxuerfPYMAzvJ7EU1u72o
/5V9zdim+hPq+9cvsqO7dkJx+cEpgwHKzrMCQQDoYqArJP/Hi34hpaiIxOdCbbtc
MY4uV8noiqDRf/pvAkyMMM+nm5p42FQZmKP8wrLCHP72/y6kIIsIIUx1sZ85AkEA
15eWR4jEoXMIFkd7Onc6tKuJ8kokqfl08DhP3mv0EmoBeFgUnGhBRxSjnIQUDwck
Bnj6KW1fk+UM29dUDjmTqQJBAITyP16eBWjlzcgaxoT2d/d7S7AXvgCxCzRRi5ph
IavMyjrhDKyBGZ0mI6eoREaC4bxl31RRkYtg9mNeU3TxsBM=
-----END RSA PRIVATE KEY-----
```

json request body:

代码块

```
{"merchantId":"bybit","clientId":"xx","paymentType":"E_COMMERCE","merchantTradeNo":"8751738a-97a4-49da-8b64-185ffcb98a7a","payId":"01JXWP4XK4ZRVYD3HT9PD52KCD","status":"PAY_SUCCESS","amount":"1.23","currency":"USDT","currencyType":"crypto","createTime":1750088344,"paymentTime":1750088344,"finishTime":1750088344,"customer":{"external_user_id":"123","user_name":"tester","register_time":"1750088344","kyc_country":"HK","uid":"foo"}}
```

1. Get the current timestamp `1751441054`
2. Combine `${timestamp}` and request body (if the http method is `POST` , the request body is `json` text, if the http method is `GET` , the request body is unescape `query` string, like `name=foo&age=18` is valid, `name%3dfoo%26age%3d18` is invalid) into a new string `$content`

> 代码块
>
> ```
> 1751441054{"merchantId":"bybit","clientId":"xx","paymentType":"E_COMMERCE","merchantTradeNo":"8751738a-97a4-49da-8b64-185ffcb98a7a","payId":"01JXWP4XK4ZRVYD3HT9PD52KCD","status":"PAY_SUCCESS","am
> ```

```
ount":"1.23","currency":"USDT","currencyType":"crypto","createTime":1750088344,"paymentTime":1750088344,"finishTime":1750088344,"customer":{"external_user_id":"123","user_name":"tester","register_time":"1750088344","kyc_country":"HK","uid":"foo"}}
```

3. Sign `$content` with `sha256` and `PKCS1V15, 1024`
4. Encode the encrypted bytes in base64

代码块

```
vOyN+NnfWppnhxS6y1D+CAllj6Z/3np1Tm+nrt16e/EDl4VZjU2sVPSS/cBcf5Hy/jBarA8Y7yrvYqJonJAsAZcFKu9twW2XWyMbURC63Iumh5gkAE9UEx4/irpX4W6KXhqc2+7wc1tapC5zfVWRMIQ5Dh+7VscxLY+/WjKb/Vw=
```

8.2 Decrypt

1. Get `timestamp` , `signature` from request header.
2. Get request body (if the http method is `POST` , the request body is `json` text, if the http method is `GET` , the request body is `unescape query` string)
3. Combine `${timestamp}` and response body into a new string `$content`
4. decode the signature bytes from `base64` .
5. Encrypt string content with `sha256` and verify signature base `PKCS1V15,1024` .

e.g. We provided the following data.

代码块

```
vOyN+NnfWppnhxS6y1D+CAllj6Z/3np1Tm+nrt16e/EDl4VZjU2sVPSS/cBcf5Hy/jBarA8Y7yrvYqJonJAsAZcFKu9twW2XWyMbURC63Iumh5gkAE9UEx4/irpX4W6KXhqc2+7wc1tapC5zfVWRMIQ5Dh+7VscxLY+/WjKb/Vw=
```

The timestamp is `1751441054` , the RSA key is

代码块

```
-----The following key pairs are for testing only-----
-----BEGIN RSA PUBLIC KEY-----
MIGJAoGBAOFSnhqtu40TOtok+yXeB+O76PXb/VAJU4Yih6hViOdSGd7imWmCSZyP
psl3TmLhUoB+rTzYDdYrlYYng6cVn6yUhxjpMWD4Qp5K4GzjvUM0f+AxlKYMj8OQ
AgDPmZG1a5ydFrje4PLytC7sUw3GP4TTk8xg6iMHmYPdRDv7AEWdAgMBAAE=
-----END RSA PUBLIC KEY-----

```

```
-----BEGIN RSA PRIVATE KEY-----
MIICXwIBAAKBgQDhUp4arbuNEzraJPsl3gfju+j12/1QCVOGIoeoVYjnUhne4plp
gkmcj6bJd05i4VKAfq082A3WK5WGJ4OnFZ+slIcY6TFg+EKeSuBs471DNH/gMZSm
DI/DkAIAz5mRtWucnRa43uDy8rQu7FMNxj+E05PMYOojB5mD3UQ7+wBFnQIDAQAB
AoGBAIqpeCi83516xw32EiSEleBdBIlA34vAoYj7ePZr5PtN+CPE2Von1rslYCO9
axzIwyfWTTATWbiCS9sqOAe78+1t3shU0ryXelWQ95LOdiMp0ZvU3ya6MzSVuiuL
6gOQYu8kzRqCzqvyMNdAHFMmP0iESefYegaiU46yQo2ZZAtVAkEA/N6gl4tYJBef
e/Gxi7qhuIKz0mvfA/yieRuiVhFb25oJaagU6hBuzegmbs45hh+n4bw8AA9k0bvi
3IBIdV/CbwJBAOQcsOPf+lrs4PEJYUQ8/eqiG7tgIUeXxuerfPYMAzvJ7EU1u72o
/5V9zdim+hPq+9cvsqO7dkJx+cEpgwHKzrMCQQDoYqArJP/Hi34hpaiIxOdCbbtc
MY4uV8noiqDRf/pvAkyMMM+nm5p42FQZmKP8wrLCHP72/y6kIIsIIUx1sZ85AkEA
15eWR4jEoXMIFkd7Onc6tKuJ8kokqfl08DhP3mv0EmoBeFgUnGhBRxSjnIQUDwck
Bnj6KW1fk+UM29dUDjmTqQJBAITyP16eBWjlzcgaxoT2d/d7S7AXvgCxCzRRi5ph
IavMyjrhDKyBGZ0mI6eoREaC4bxl31RRkYtg9mNeU3TxsBM=
-----END RSA PRIVATE KEY-----
```

1. Get `timestamp` , `signature` from request header.

> `timestamp` = `1751441054`
>
> `signature` = `vOyN+NnfWppnhxS6y1D+CAllj6Z/3np1Tm+nrt16e/EDl4VZjU2sVPSS/cBcf5Hy/jBarA8Y7yrvYqJonJAsAZcFKu9twW2XWyMbURC63Iumh5gkAE9UEx4/irpX4W6KXhqc2+7wc1tapC5zfVWRMIQ5Dh+7VscxLY+/WjKb/Vw=`

2. Get request body (if the http method is `POST` , the request body is `json` text, if the http method is `GET` , the request body is `unescape query` string)

> 代码块
>
> ```
> {"merchantId":"bybit","clientId":"xx","paymentType":"E_COMMERCE","merchantTradeNo":"8751738a-97a4-49da-8b64-185ffcb98a7a","payId":"01JXWP4XK4ZRVYD3HT9PD52KCD","status":"PAY_SUCCESS","amount":"1.23","currency":"USDT","currencyType":"crypto","createTime":1750088344,"paymentTime":1750088344,"finishTime":1750088344,"customer":{"external_user_id":"123","user_name":"tester","register_time":"1750088344","kyc_country":"HK","uid":"foo"}}
> ```

3. Combine `${timestamp}` and response body into a new string `$content`

> 代码块
>
> ```
> 1751441054{"merchantId":"bybit","clientId":"xx","paymentType":"E_COMMERCE","merchantTradeNo":"8751738a-97a4-49da-8b64-185ffcb98a7a","payId":"01JXWP4XK4ZRVYD3HT9PD52KCD","status":"PAY_SUCCESS","amount":"1.23","currency":"USDT","currencyType":"crypto","createTime":1750088344,"paymentTime":1750088344,"finishTime":1750088344,"customer":
> ```

```
{"external_user_id":"123","user_name":"tester","register_time":"1750088344","
kyc_country":"HK","uid":"foo"}}
```

4. Decode the signature bytes from base64
5. Encrypt string content with sha256 and verify signature base PKCS1V15,1024

10. Error Code

ErrorCode	Description	Solution
100000	success	
400000	Invalid request parameters	Verify the format of the requested data.
400002	Signature verification failed	Ensure the merchant signature is correct.
400003	Request timestamp timeout	Verify the timestamp field in the request header.
400004	Order not found	Pay(Refund) order not found
400007	Unsupported payment type	Confirm that the payment type is supported by the API.
400008	Invalid precision	Confirm that the precision is in range.
400009	Unsupported currency	Confirm that the currency is supported.
400010	Quotation failed	Fx convert quotation failed.
400603	Order timeout	Check if the order has expired.
400621	Incorrect payment amount	Verify the requested amount.
400620	Duplicate payment for order	Ensure the merchant order number has not been submitted twice.
500000	Bybit internal error	Bybit internal error, such as db error.
500008	Merchant not found	Confirm that the provided merchant ID is correct.
500100	Payment QR code has expired	Place a new order to generate a new QR code.
500101	Duplicate payment for QR code	Verify the order status before proceeding.
500102	Merchant not Available	Merchant not Available by API now

500103	Order has refund	Order has refund, do not refund again
500104	Balance not Available	Balance not available, top up to the refund account(Bybit KYB funding account)
500105	Order not paid	Order not paid
500106	Version not supported	The features are not supported in the old version, pls upgrade the api version.
600001	AUTHORIZE_CODE_INVALID	Invalid authorization code
600002	AUTHORIZE_CODE_EXPIRED	Authorization code expired
600003	AUTHORIZE_TOKEN_INVALID	Access token invalid or expired
600004	AUTHORIZE_TOKEN_EXPIRED	Access token expired
600005	AUTHORIZE_SCOPE_INVALID	Invalid authorization scope
600006	AUTHORIZE_REJECTED	User rejected authorization