



# **DOLPHIN 2019**

## **FICHE TECHNIQUE**



**UN SERIOUS GAME COMPETITIF DE ROBOT ADVISOR**

# TABLE DES MATIERES

I.	API REST JSON pour le projet Dolphin .....	3
A.	Schéma des requêtes.....	3
B.	Authentification .....	3
1)	Exemple en JAVA : .....	4
2)	Exemple en Python .....	4
II.	Exemple de méthodes disponibles .....	5
3)	Récupération de la liste d'actifs en base.....	5
4)	Récupération des valeurs de cotation (get asset/<id>/quote).....	6
5)	Récupération de la liste des ratios.....	7
6)	Récupération de la composition d'un portefeuille.....	8
7)	MAJ de la composition d'un portefeuille .....	9
8)	Calcul de ratios.....	9
9)	Conversion de taux .....	10
C.	Ratios mis à disposition .....	12
1)	Beta (Identifiant : 7).....	13
2)	Corrélation (Identifiant : 11) - Corrélation de Pearson .....	14
3)	Rendement (Identifiant : 13) .....	15
4)	Rendement annualisé (Identifiant : 9) .....	16
5)	Sharpe (Identifiant : 12) .....	17
6)	VaR historique (Identifiant : 14) .....	18
7)	Volatilité (Identifiant : 10) .....	20
8)	Exposition action (Identifiant : 15).....	21

## I. API REST JSON pour le projet Dolphin

JUMP met à disposition pour le projet Dolphin un serveur de test accessible via une API REST JSON.

*« Le protocole REST (Representational State Transfer) constitue un style architectural et un mode de communication fréquemment utilisé dans le développement de services Web. »*

L'API REST JSON permet d'accéder au serveur d'application JUMP de test via des requêtes http. Les objets transférés sont des objets JSON.

### A. Schéma des requêtes

Comme dit plus haut, l'API est utilisable via des requêtes http. Ici nous allons énumérer quelques exemples de requêtes utilisables. La liste complète est consultable dans le fichier « *documentation\_JUMP\_API\_REST.html* » fourni.

Les résultats des requêtes sont retournés sous la forme d'objet JSON. Il existe de multiples moyens de parser ce type d'objet. De plus, un exemple concret de code Java est aussi fourni, dont vous pouvez vous inspirer.

Le schéma initial de nos requêtes **https://IP:PORT/api/v1/**

### B. Authentification

Tous les services REST utilisent l'authentification [HTTP Basic](#). Chaque requête HTTP émise contient les informations d'authentification de l'utilisateur.

## 1) Exemple en JAVA :

```
// 1. Create AuthCache instance
AuthCache authCache = new BasicAuthCache();

// 2. Generate BASIC scheme object and add it to the local auth cache
BasicScheme basicAuth = new BasicScheme();
final HttpHost locLocalhost = new HttpHost(parHostName, parPort, SCHEME);
authCache.put(locLocalhost, basicAuth);

// 3. Add AuthCache to the execution context
_clientContext = HttpClientContext.create();
_clientContext.setAuthCache(authCache);

CredentialsProvider credsProvider = new BasicCredentialsProvider();
credsProvider.setCredentials(
    new AuthScope(locLocalhost),
    new UsernamePasswordCredentials(parUsername, parPassword));
```

## 2) Exemple en Python

```
URL = 'https://hostname:port/api/v1/'
AUTH = ('username', 'password')

def get_data(endpointApi, date=None, full_response=False, columns=list()):
    payload = {'date': date, 'fullResponse': full_response }
    res = requests.get(URL + endpointApi + columns_to_str(columns),
                       params=payload,
                       auth=AUTH,
                       verify=False)
    return res.content.decode('utf-8')
```

## II. Exemple de méthodes disponibles

### 3) Récupération de la liste d'actifs en base

GET

SCHEMA/asset?columns=ASSET\_DATABASE\_ID&columns=LABEL&columns=TYPE&columns=LAST\_CLOSE\_VALUE\_IN\_CURR&date=2013-06-14

Cette requête GET permet de récupérer une liste des actifs avec les informations ci-dessous :

- ASSET\_DATABASE\_ID : Identifiant de l'actif en base.
- LABEL : Nom de l'actif.
- TYPE : La type de l'actif.
- LAST\_CLOSE\_VALUE\_IN\_CURR : La valeur de clôture de l'actif.
- Date : La date à utiliser pour ces colonnes.

La requête va alors retourner une la liste des actifs avec les informations demandées, sous format JSON.

Résultat :

```
[{
  "LAST_CLOSE_VALUE_IN_CURR": {
    "type": "currency_value",
    "value": "56,23USD"
  },
  "LABEL": {
    "type": "string",
    "value": "ABBOTTLABS"
  },
  "ASSET_DATABASE_ID": {
    "type": "int32",
    "value": "315"
  },
  "TYPE": {
    "type": "asset_type",
    "value": "STOCK"
  }
},
... ]
```

On remarquera que la plupart des valeurs retournées sont composées de deux Strings : un pour le type et un autre pour la valeur.

Le document de référence de l'API en annexe de ce document permet de retrouver l'ensemble des types disponibles, « *enum\_jump\_value\_type* ».

#### 4) Récupération des valeurs de cotation (get asset/<id>/quote)

GET

SCHEMA/asset/59/quote?start\_date=2013-06-14&end\_date=2019-05-31

Cette requête GET permet de récupérer la liste des cotations dans la période du 15/06/2013 au 31/05/2019

Résultat :

```
[  
  {  
    "date": "2013-06-14",  
    "nav": 189.91,  
    "gross": 189.91,  
    "pl": 0.6999999999999886,  
    "close": 189.91,  
    "return": 0.0036995930447650156  
  },...  
]
```

Le résultat est une liste d'objet avec comme information :

- La date des valeurs de cotations
- La NAV de l'actif à la date
- Le montant brut de l'actif à la date
- La plus ou moins-value de l'actif à la date
- La valeur de clôture de l'actif à cette date
- Le rendement de l'actif à cette date

## 5) Récupération de la liste des ratios

GET

SCHEMA/ratio

Cette requête GET permet de récupérer la liste des ratios se trouvant en base.

Résultat :

```
[{u'id': 7,  
  u'is_benchmark_needed': True,  
  u'is_percent': False,  
  u'name': u'Beta',  
  u'type': u'Ratio'},  
...  
]
```

Comme on peut le voir, un ratio est composé :

- D'un identifiant unique
- D'un type
- D'un nom
- D'un booléen pour définir s'il a besoin d'un benchmark.  
Un benchmark est un actif utilisé comme référence dans un calcul
- D'un booléen pour savoir le type de valeur de retour du ratio

## 6) Récupération de la composition d'un portefeuille

GET

SCHEMA/portfolio/564/dyn\_amount\_compo

Cette requête GET permet de récupérer la composition du portefeuille d'identifiant 564.

Résultat :

```
[
  {
    "label": "EPITA_PTF_1",
    "currency": {
      "code": "EUR"
    },
    "type": "front",
    "values": {
      "2013-06-14": [
        {
          "asset": {
            "asset": 42,
            "quantity": 2
          }
        }
      ]
    }
  }
]
```

Un portefeuille est donc composé :

- D'un label
- D'une devise. Voici la liste des devises actuellement dans la base  
{"EUR","GBp","JPY","NOK","SEK","USD"}
- D'un type de portefeuille Jump. **Vous ne devez utiliser que les portefeuilles « front ».** Sinon les calculs de ratios **sur vos portefeuilles ne fonctionneront pas comme prévus.**
- D'une map avec :
  - La date de la composition. Pour rappel votre composition doit être **unique** et doit **dater du 2013-06-14**.
  - Une liste des containers de lignes. Dans notre cas une ligne d'actif avec comme actif cible l'actif 42 et comme quantité 2.



## 7) MAJ de la composition d'un portefeuille

PUT

SCHEMA/portfolio/564/dyn\_amount\_compo

Avec comme entité un portefeuille respectant le modèle d'un portefeuille Json. L'entité à utiliser est rappelée dans la partie Récupération de la composition d'un portefeuille ou dans le documentation JUMP API REST.html

Résultat :

Ecrase le contenu du portefeuille 564 par celui de l'entité.

## 8) Calcul de ratios

POST

SCHEMA/ratio/invoke

Payload (post body):

```
{
  _ratio=[9, 12, 18],
  _asset=[564],
  _bench=null,
  _startDate=2013-06-14,
  _endDate=2019-05-31,
  _frequency=null
}
```

Avec une entité RatioMultiAssetParamModel décrite de la manière suivante :

- **\_ratio** : Une liste d'entier représentant les identifiants numériques de nos ratios
- **\_asset** : Une liste d'entier représentant les identifiants numériques des actifs dont on veut calculer les ratios
- **\_bench** : Un entier représentant l'identifiant numérique d'un actif utilisé comme référence dans un calcul. Cette valeur est optionnelle.
- **\_startDate** : Date de début de la période
- **\_endDate** : Date de fin de la période
- **\_frequency** : représente la fréquence des données que nous allons utiliser pour nos calculs. Une valeur « *null* » serait traitée comme une fréquence journalière.

L'exemple ci-dessous est un RatioMultiAssetParamModel permettant de calculer les ratios de rendement annualisé (id : 9), de Sharpe (id : 12) et de la volatilité (id : 10) sur le portefeuille d'identifiant 564 dans la période du 14/06/2013 au 31/05/2019.

**Résultat :**

```
{
  "564": {
    "9": {
      "type": "percent",
      "value": "0,140714359885"
    },
    "10": {
      "type": "percent",
      "value": "0,299422561689"
    },
    "12": {
      "type": "double",
      "value": "0,453253619631"
    }
  }
}
```

Le résultat est sous la forme d'une map avec :

- Une clef représentant l'identifiant de l'actif
- Une map représentant les résultats des ratios avec comme clef l'identifiant du ratio cible et comme valeur la valeur du ratio pour l'actif.

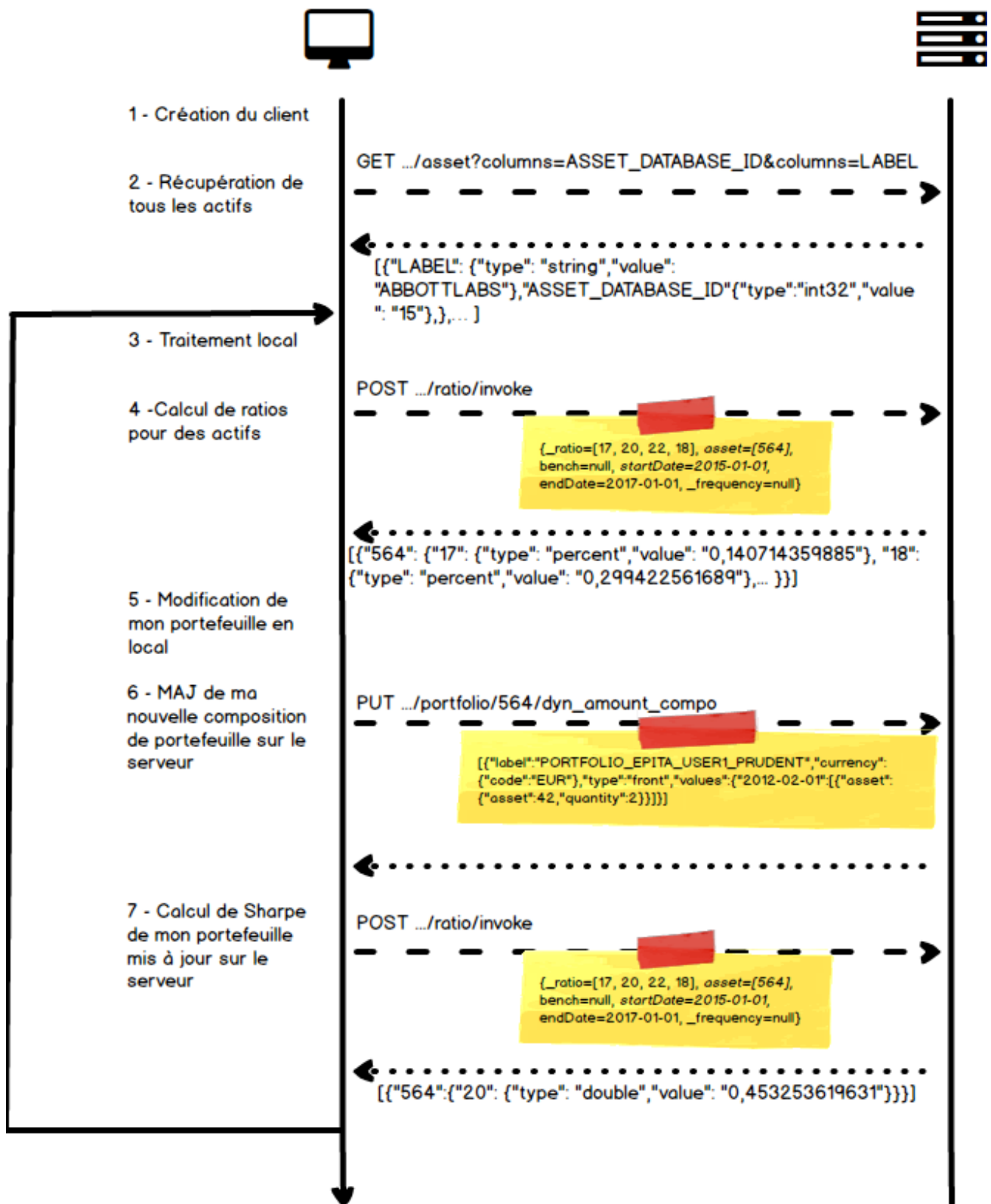
## 9) Conversion de taux

GET

SCHEMA /currency/rate/<currency\_src>/to/<currency\_dest>

Exemple :

GET /currency/rate/GBP/to/EUR



## **C. Ratios mis à disposition**

Voici la liste des ratios que nous vous mettons à disposition dans le cadre du projet Dolphin.

Il ne s'agit que d'un tout petit sous-ensemble des centaines de ratios disponibles dans le module de risque et de performance du progiciel JUMP.

Dans le cadre des appels REST, vous pourrez les appeler par leurs identifiants.

## 1) Beta (Identifiant : 7)

### Définition

---

Le Beta mesure la volatilité d'un actif et se définit comme le rapport de la covariance des rentabilités du portefeuille avec celles de l'indice du benchmark et de la variance des rentabilités du benchmark. Le Beta est aussi le rapport entre la rentabilité de cet actif et celle du marché puisque la volatilité concerne les variations de cours qui sont un élément essentiel de rentabilité.

### Formule

---

Générique :  $Beta(X, Y) = \frac{covariance(X, Y)}{variance(Y)}$

### Interprétation

---

Par exemple, si le Beta d'une action est de 0.8, son cours a varié en moyenne dans la période précédente de 0,8% quand le marché variait de 1%. S'il est de 1.5, il a varié de 1.5%. Autrement dit le Beta est la sensibilité ou l'élasticité du cours de l'actif par rapport à l'indice boursier représentant le benchmark.

### Données

---

- ☐ Nécessite un benchmark
- ☐ Type de retour : double

### Paramètres dans l'implementation fournie

---

- ☐ Variance calculée en mode période (varA dans Excel)

## 2) Corrélation (Identifiant : 11) - Corrélation de Pearson

### Définition

Corrélation de Pearson sur les rendements.

### Formule

$$Correlation_{Pearson}(a, b) = \frac{Covariance(R_a, R_b)}{\sqrt{Variance(R_a) * Variance(R_b)}}$$

$$Covariance(X, Y) = \frac{1}{N} \sum_{i=1}^{i=n} (x - \bar{X})(y - \bar{Y})$$

### Interprétation

Une corrélation extrême ( $P = 1$  ou  $-1$ ) signifie une dépendance linéaire entre le fonds et le benchmark. On s'attendra alors à voir les variables évoluer proportionnellement dans la même direction si est  $P$  positif, dans des directions opposées sinon.

Il est rare, hors de problèmes de démonstration, de constater une corrélation aussi élevée. On admet habituellement une relation forte entre les variables à partir de  $P \pm 0.9$ .

Si la corrélation est nulle, il n'y aura pas de proportionnalité entre les évolutions de  $X$  et  $Y$ .

### Données

- ☐ Utilise les rendements
- ☐ Nécessite un benchmark
- ☐ Nécessite des variances (des Rendements) non-nulles
- ☐ Type de retour : double

### Paramètres dans l'implémentation fournie

- ☐ Variance calculée en mode période (varA dans Excel)

### 3) Rendement (Identifiant : 13)

#### Définition

---

Calcule le rendement d'un actif ou d'un benchmark. Il existe deux types de rendements, les rendements arithmétiques et les rendements logarithmiques. Ce ratio peut également renvoyer le résultat annualisé.

#### Formule

---

$$\text{Rendement } NAV_a = \frac{NAV_i - NAV_{i-1}}{NAV_{i-1}}$$

#### Interprétation

---

Permet d'indiquer si l'actif a gagné de la valeur ou en a perdu entre deux cotations.

#### Données

---

☐ Type de retour : pourcentage

#### Paramètres dans l'implementation fournie

---

☐ Annualiser : Non

## 4) Rendement annualisé (Identifiant : 9)

### Définition

---

Dans le calcul de certains ratios, les rendements élémentaires sont normalisés. La normalisation des rendements permet de passer de l'observation des performances sur une période quelconque à l'estimation de ce qu'aurait été cette performance sur une période donnée, différente de la première.

Dans le cadre du projet Dolphin, nous vous mettons à disposition un rendement annualisé.

### Formule

---

Méthode de normalisation

$$(1 + \text{Rendement})^{\text{NbPeriodeNbJours}} - 1$$

Où *NbPeriode* = 365 pour la fréquence journalière.

Et *NbJours* = nombres de jours de la période étudiée.

### Interprétation

---

Permet d'indiquer si l'actif a gagné de la valeur ou en a perdu entre deux cotations, le tout normalisé sur une année.

### Données

---

☐ Type de retour : pourcentage

### Paramètres dans l'implémentation fournie

---

☐ Annualiser : Oui



## 5) Sharpe (Identifiant : 12)

### Définition

---

Le ratio de Sharpe mesure l'écart de rentabilité d'un portefeuille d'actifs financiers (actions par exemple) par rapport au taux de rendement d'un placement sans risque (autrement dit la prime de risque, positive ou négative), divisé par un indicateur de risque, l'Ecart Type de la rentabilité de ce portefeuille (autrement dit sa volatilité).

L'implémentation retenue pour le projet Dolphin est une variante assez basique avec un taux sans risque fixe.

### Formule

---

$$S(a) = \frac{\langle R \rangle - R_f}{\sigma_T}$$

$\langle R \rangle$  : Taux de retour sur investissement du portefeuille considéré :

$$\langle R \rangle = \left( 1 + \frac{X_a \text{Fin} - X_a \text{Debut}}{X_a \text{Debut}} \right)^{\frac{365}{nbDay}} - 1$$

$\sigma_T$  : le taux de retour sur investissement du portefeuille considéré.

$R_f$  : Référentiel de comparaison choisi (souvent le taux de placement sans risque).

### Interprétation

---

Ce ratio permet de répondre à la question suivante : le gestionnaire parvient-il à obtenir un rendement supérieur avec davantage de risque ?

Intuitivement on peut l'interpréter comme l'excès des rendements par unité de risque.

### Données

---

☐ Type de retour : double

### Paramètres dans l'implémentation fournie

---

☐  $R_f = 0,05\%$

☐ Variance calculée en mode période (varA dans Excel)

## 6) VaR historique (Identifiant : 14)

### Définition

La VaR (Value at Risk) est une notion utilisée généralement pour mesurer le risque de marché d'un portefeuille d'instruments financiers. Elle correspond à la pire perte possible en fonction d'une probabilité et d'un horizon temporel donnés, son horizon étant défini en fonction de la fréquence.

La VaR historique se base uniquement sur l'interprétation des rendements passés.

### Formule

$$VaR(a) = \text{quantile}(R_a, \text{Threshold})$$

Quantile: formule MS Excel : CENTILE (Matrice des rendements ; 1- taux)

Threshold : seuil paramétré (compris entre [0.0; 0.9999...])

### Interprétation

En fréquence journalière, le résultat correspond à la pire perte possible dans X% des cas, X étant le seuil, en un jour.

### Données

☐ Type de retour : pourcentage

### Extension

Par défaut l'horizon de la VaR dans JUMP est de 1 jour. Il est cependant aisé de modifier l'horizon de la VaR en utilisant une approximation classique.

En effet

$$VaR(10j, X\%) \approx \sqrt{10} \times VaR(1j, X\%)$$

### Remarque :

La VaR(10j, 99%) a des valeurs très proches de la VaR(20j, 95%) pour les raisons suivantes :

Une autre approximation usuelle donne :

$$VaR(T, X') \approx \frac{N^{-1}(X')}{N^{-1}(X)} \times VaR(T, X)$$

Si l'on pose  $\frac{N^{-1}(0.99)}{N^{-1}(0.95)} \approx 1.4143$

$$\sqrt{10} \approx 1.4142$$

On a

$$\begin{aligned}\text{VaR}(10j, 99\%) &\approx \frac{N^{-1}(0.99)}{N^{-1}(0.95)} \times \text{VaR}(10j, 95\%) \approx \sqrt{10} \times \text{VaR}(10j, 95\%) \approx \text{VaR}(10j + 10j, 95\%) \\ &\approx \text{VaR}(20j, 95\%)\end{aligned}$$

En raison des approximations et du nombre de chiffres significatifs, on aura donc des valeurs identiques à la deuxième décimale pour la  $\text{VaR}(10j, 99\%)$  et la  $\text{VaR}(20j, 95\%)$ .

### Paramètres dans l'implémentation fournie

---

☐  $X = 95\%$

### Liens

---

☐ <http://www.itl.nist.gov/div898/handbook/prc/section2/prc252.htm>

☐ <http://en.wikipedia.org/wiki/Quantile>

☐ [http://jb.desquilbet.pagesperso-orange.fr/docs/A\\_M2\\_GRM\\_3VaR.pdf](http://jb.desquilbet.pagesperso-orange.fr/docs/A_M2_GRM_3VaR.pdf)

## 7) Volatilité (Identifiant : 10)

### Définition

---

La volatilité est une mesure de l'instabilité du cours d'un actif financier. Elle sert de paramètre de quantification du risque de rendement et de prix d'un actif financier. Elle représente globalement un écart type annualisé.

### Formule

---

Générique :  $Volatilité(X, nbPeriodPerYear) = sdev(X) * \sqrt{nbPeriodPerYear}$

Application :  $\sigma T(a) = \sigma(a) \sqrt{T}$

T : L'horizon de temps : Méthode de normalisation

### Données

---

☐ Type de retour : pourcentage

### Paramètres dans implementation fournie

---

☐ Variance calculée en mode période (varA dans Excel)

## 8) Exposition action (Identifiant : 15)

### Définition

---

Calcule l'exposition Action d'un portefeuille. C'est-à-dire :  
(Montant des lignes présentant une exposition action) / (montant total du portefeuille)

### Grille de classification

---

Dans cette implémentation simplifiée pour le projet Dolphin, les actifs comptabilisés dans l'exposition action sont :

- ✓ Les actifs de type Action
- ✓ Les fonds présentant comme valeur de l'attribut « focus sur une classe d'actif » les valeurs « Action » ou « Alternatif »

### Données

---

☐ Type de retour : pourcentage de l'exposition action du portefeuille