

Composant Serveur API REST pour WinDev®

Léger.

Rapide.

Gratuit & Open Source.

Sans serveur IIS/Apache/...

Sans licence Serveur WebDev®.

Pas de SAAS, pas d'abonnement.

Windows 32/64 & Linux

Présentation

NEW : LightREST V3

La Documentation



Téléchargements

Un coup de main ?

Classe IrRequest



Role

Objet reçu par les méthodes REST associées aux routes.

Contient la requête HTTP reçue par LightREST : headers, URL, variables de route, body, fichiers, données custom, et structures d'authentification.



Membres et propriétés

Nom	Type	Usage
Authentication	Structure <i>IrAuthentication</i>	Infos d'authentification reçues dans la requête REST
Body	Buffer	Body de la requête REST. Contenu au format texte, binaire, JSON, XML, ...
ContentLength	entier sur 8 octets	Longueur du body en octets
CustomData	Variant	Données custom précédemment attachées à la requête par un <i>HOOK</i> .

Voir également

[IrRequest:ServerCustomData](#) et

[IrRequest:RouteCustomData](#).

Fil
es

ta
ble
au
as
so
cia
tif
de
Str
uc
tur
es
[IrFile](#)

Tableau des fichiers
uploadés

Hea
der

ta
ble
au
as
so
cia
tif
(c
cS
an
sC
as
se

Headers HTTP

	+A ve cD ou bl on) de ch aî ne	
Host	chaîne	Hôte
ID	chaîne	Identifiant LightREST interne de la requête. Utile par exemple pour analyser les logs LightREST
Method	chaîne	Méthode HTTP (GET/POST/UPDATE...) .
PostForm	tableau associatif (c	Champs de formulaire POST

	cS an sC as se) de ch aî ne	
Pro to	ch aî ne	Protocole (ex: HTTP/1.1).
Rem ote Add r	ch aî ne	Adresse distante (IP:port client)
Req ues tUR I	ch aî ne	URI brute.
Rou teC ust omD ata	Va ria nt	Données custom attachées lors de la création de la route <u>IrRoute:CustomData.</u>
Ser ver	Va ria	Données custom attachées au serveur

Custom Data	nt	<u>IrServer.CustomData.</u>
URL	Structure <u>IrU</u> <u>r/</u>	Composition de l'URL
Vars	tableau accessible sovia tif (c cS an sC as se) de ch a ne	Variables extraites de la route ({id} → Vars["id\"]). Accessibles avec la méthode :GetRouteValue().

  **Types**

 

IrURL — Décomposition de l'URL de la requête.

 

IrFile — Décrit un fichier uploadé dans la requête.

 

IrAuthentication — Informations d'authentification fournies dans la requête.

Méthodes

  **GetHeaderValue**

 **Présentation**

Renvoie la valeur d'un header HTTP

 **Prototypage**

`GetHeaderValue(pHeader chaîne)`

 **Paramètres**

No	T	Usage
m	y	

**p
e**

pH	c	Nom du header.
ea	h	
de	aî	
r	n	
	e	

Valeurs retournées

Ty pe	Usage
ch aîn e	Valeur du HEADER HTTP reçu (chaîne vide si absente)

Exemple

- Lecture d'une clé d'API transmise dans le header *X-API-KEY*
- Refus en StatusUnauthorized si la clé est absente ou invalide
- Exemple typique : sécuriser une route de "ping" ou un endpoint d'administration

```
FONCTION  
PingSecure(poRequest objet  
lrRequest, poResponse objet
```

Copier

```
lrResponse)

sApiKey est chaîne

sApiKey =
  poRequest:GetHeaderValue("X-
  API-KEY")

SI sApiKey="" OU
sApiKey<>"MY_SECRET_KEY"
ALORS
  poResponse:Status      =
  lrResponse::StatusUnauthoriz
  ed
  poResponse:ContentType =
  lrResponse::ContentTXT
  poResponse:Body        =
  "Unauthorized"
SINON
  poResponse:Status      =
  lrResponse::StatusOK
  poResponse:ContentType =
  lrResponse::ContentTXT
  poResponse:Body        =
  "pong"
FIN
```

GetHeaderSubValue

Présentation

Extrait une sous-valeur dans un header HTTP multi-valeur (exemples : boundary= ou cache-control=).

Prototypage

```
GetHeaderSubValue (pHeader chaîne
```

Paramètres

No m	T y p e	Usage
pH ea de r	c h aîne	Nom du header (ex: Content-Type).
pS ub He ad er	c h aîne	Sous-header (ex: boundary).

Valeurs retournées

Ty pe	Usage
ch aîne	Valeur du SUB HEADER HTTP (Chaîne vide si absent)

 Exemple

- Lecture de la sous-valeur *boundary* d'un header *Content-Type* de type *multipart/form-data*
- Retour `StatusBadRequest` si la sous-valeur est absente (requête invalide / upload mal formé)
- Utile pour diagnostiquer rapidement un problème d'upload (par ex. depuis Postman / navigateur)

Copier

```
FONCTION
UploadPOST(poRequest objet
lrRequest, poResponse objet
lrResponse)

sBoundary est chaîne

// Exemple attendu côté
client :
// Content-Type:
multipart/form-data;
boundary=----
WebKitFormBoundary7MA4YWxkTr
Zu0gW

sBoundary =
poRequest:GetHeaderSubValue (
"Content-Type", "boundary")

SI sBoundary="" ALORS
    poResponse:Status =
lrResponse::StatusBadRequest
    poResponse:ContentType =
lrResponse::ContentTXT
    poResponse:Body =
"Missing boundary in
Content-Type header"
```

```
SINON
    poResponse:Status      =
    lrResponse::StatusOK
    poResponse:ContentType =
    lrResponse::ContentTXT
    poResponse:Body        =
    "Boundary detected: " +
    sBoundary
FIN
```

▼ (X) GetRouteValue

📖 Présentation

Renvoie la valeur d'une variable extraite de la route.

Par exemple, pour la route */tiers/123456*, *GetRouteValue("id")* retournera *123456*.

🧩 Prototypage

```
GetRouteValue(pTag chaîne) : ch
```

🔧 Paramètres

No m	T y p e	Usage
---------	------------------	-------

pT	c	Nom de la variable
ag	h	(ex: id).
	aî	
	n	
	e	

Valeurs retournées

Ty pe	Usage
ch aîn e	Valeur de la variable de ROUTE (chaîne vide si absente)

Exemple

- Route de type */tiers/{id}*
- Extraction de l'ID à partir de la route (variable {id})
- Recherche de l'enregistrement
- Conversion en variant avec la méthode IrResponse::RecordToVariant puis en JSON
- envoi de la réponse (ou *StatusNotFound* si absent)

```
FONCTION TiersGET(poReq  
objet lrRequest, poResponse  
objet lrResponse)
```

[Copier](#)

```
sID      est chaîne
vTiers est Variant

// Exemple de route appelée
: /tiers/123456
sID =
poRequest:GetRouteValue("id"
)

SI sID="" ALORS
    // Normalement impossible
    si la route est bien
    déclarée, mais sécurise le
    code
    poResponse:Status      =
    lrResponse::StatusBadRequest
    poResponse:ContentType =
    lrResponse::ContentTXT
    poResponse:Body        =
    "No ID provided"
    RETOUR
FIN

SI
hLitRecherchePremier("TIERS"
, "ID", sID) ALORS
    vTiers =
    lrResponse::RecordToVariant(
    "TIERS", "ID,NAME,CITY")
    poResponse:Body      =
    VariantVersJSON(vTiers)
    poResponse:ContentType =
    lrResponse::ContentJSON
    poResponse:Status      =
    lrResponse::StatusOK
SINON
    poResponse:Body      =
    "Not found"
    poResponse:ContentType =
    lrResponse::ContentTXT
    poResponse:Status      =
    lrResponse::StatusNotFound
FIN
```

(X)GetURLValue

Présentation

Renvoie la valeur d'un paramètre situé dans l'URL.

Exemple : Pour l'URL */tiers/123456?scope=full&country=FR*,
GetURLValue("country") retournera *FR*.

Prototypage

```
GetURLValue(pTag chaîne) : chaî
```

Paramètres

No m	T y p e	Usage
pT ag	c h aî n e	Nom du paramètre (ex: scope, country, q...).

Valeurs retournées

Type	Usage
------	-------

chaîne	Valeur du paramètre d'URL (chaîne vide si la variable n'existe pas)
--------	---

Exemple

- Lecture de paramètres dans l'URL (query string)
- Exemple : */tiers/123456?scope=full&country=FR*
- Application d'une valeur par défaut si le paramètre n'est pas fourni

```
FONCTION
TiersGET_Scope(poRequest
objet lrRequest, poResponse
objet lrResponse)

sScope est chaîne
sCountry est chaîne

// Exemple appelé :
/tiers/123456?
scope=full&country=FR
sScope =
  poRequest:GetURLValue("scope")
sCountry =
  poRequest:GetURLValue("country")

SI sScope="" ALORS
  sScope = "basic" //
valeur par défaut
```

[Copier](#)

FIN

```
poResponse:Status      =  
lrResponse::StatusOK  
poResponse:ContentType =  
lrResponse::ContentTXT  
poResponse:Body        =  
"scope=" + sScope + ",  
country=" + sCountry
```



VariantToSource



Présentation

Copie les membres d'un variant vers un enregistrement (source de données).
Utile pour copier directement un flux reçu dans une requête REST dans un enregistrement avant insertion ou mise à jour.
Vérifie l'existence de chaque membre dans la structure de la table avant affectation.



Prototypage

```
VariantToSource (pVariant, pSour
```



Paramètres

No

T

Usage

m	y p e	
pV ar ia nt	V a ri a nt	Données à copier.
pS ou rc e	S o u rc e d e D o n n é e s	Source de données HFSQL cible.
pI DC ol um n	c h aî n e	Rubrique ID à ignorer (optionnel).

 Valeurs retournées

(aucune)

Exemple

Endpoint d'import : réception d'un JSON
les données d'un CLIENT :

- Conversion du JSON en Variant
- Affectation des membres de l'enregistrement CLIENT à partir des membres homonymes du variant
- Insertion en base HFSQL

```
FONCTION
ClientPOST(poRequest
objet lrRequest,
poResponse objet
lrResponse)

sJSON est chaîne
vData est Variant

// Exemple de body JSON
attendu :
//
{"NAME":"Dupont","FIRSTNAME":"Paul","CITY":"Paris"}

QUAND EXCEPTION DANS
    vData =
JSONVersVariant(sJSON)
FAIRE
    poResponse:Status
=
lrResponse::StatusBadRequest

poResponse:ContentType =
lrResponse::ContentTXT
```

Copier

```
        poResponse:Body
        = "Malformed JSON"
        RETOUR
    FIN

    // Copie des données vers
    un enregistrement CLIENT
    // "ID" est ignoré (si ID
    auto par exemple)
    poRequest:VariantToSource
    (vData, CLIENT, "ID")

    //Ici on peut vérifier la
    validité des données
    avant de les insérer

    hAjoute(CLIENT)

    poResponse:Status      =
    lrResponse::StatusCreated
    poResponse:ContentType =
    lrResponse::ContentTXT
    poResponse:Body        =
    "Import done"
```