

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 IrResponse



Role

Définit la réponse HTTP renvoyée par le Handler REST : code statut, content-type, headers et body



Membres et propriétés

Nom	Type	Usage

Body	Buffer	Contenu de la réponse (texte ou binaire selon IsBinary). Le format doit être cohérent avec le membre :ContentType
ContentType	chaîne	Indique le type MIME du contenu envoyé dans le membre :Body Contient généralement une des constantes Content de la classe (exemples : ContentJSON, ContentJPG, ContentZIP, ...)  Si ce membre n'est pas renseigné, le moteur LightREST indiquera <u>ContentJSON</u> par défaut, et enverra un événement de type <u>EVE_WARNING</u> (si un <u>Hook</u> a été déclaré)..
Headers	tableau	Headers HTTP de la réponse. Tableau contenant les HEADERS ajoutés avec la méthode

	tif (c cS an sC as se +A ve cD ou bl on) de ch aî ne s	:SetHeaderValue(), mais aussi ceux définis globalement au niveau du serveur LightREST avec les méthodes <u>IrServer:SetGlobalHeader()</u> et <u>IrServer:SetGlobalHeaders()</u>
IsBinary	boolean	Indique que le membre :Body contient du binaire. Ce membre est positionné automatiquement à vrai lors de l'appel à la méthode <u>:SetBinaryContent()</u>
Status	entier	Status HTTP de la réponse. Contient

habituellement une des constantes :Status*.
Si ce membre n'est pas renseigné, le moteur LightREST indiquera StatusOK* (=200) par défaut.

▼ C Constantes

Constantes MIME (valeurs possibles de la propriété :ContentType)

Nom	Type	Usage
ContentJSON	chaîne	Type MIME JSON (application/json)
ContentTXT	chaîne	Type MIME texte (text/plain)
ContentHTML	chaîne	Type MIME HTML (text/html)

ML		
ContentXML	change	Type MIME XML (application/xml)
ContentCSV	change	Type MIME CSV (text/csv)
ContentPDF	change	Type MIME PDF (application/pdf)
ContentPNG	change	Type MIME image PNG (image/png)
ContentJPEG	change	Type MIME image JPEG (image/jpeg)
ContentGIF	change	Type MIME image GIF (image/gif)

Content- Type	image/ webp	Type MIME image WebP (image/webp)
Content- Type	image/ svg+xml	Type MIME SVG (image/svg+xml)
Content- Type	application/ zip	Type MIME ZIP (application/zip)
Content- Type	application/ octet-stream	Type MIME binaire générique (application/octet- stream)

Statuts HTTP courants (valeurs possibles de la propriété :Status)

Nom	Type	Usage
Status OK	entier	<u>200</u> - Requête traitée avec succès.
Status	entier	<u>201</u> - Ressource créée.

tus Cre ate d	tie r	
Sta tus Acc ept ed	en tie r	<u>202</u> - Requête acceptée (traitement asynchrone possible, succès non garanti).
Sta tus Non Aut hor ita tiv eIn for mat ion	en tie r	<u>203</u> - La requête a réussi, mais un mandataire a modifié les en-têtes ou le contenu inclus de la réponse 200 (OK) du serveur d'origine.
Sta tus NoC ont ent	en tie r	<u>204</u> - Succès sans contenu.
Sta tus Res	en tie r	<u>205</u> - La requête a été traitée avec succès et que le client doit réinitialiser l'affichage

etC ont ent		du document.
Sta tus Par tia lCo nte nt	en tie r	<u>206</u> - Contenu partiel (Range).
Sta tus	en tie r	<u>300</u> - Requête prise en compte par le serveur mais ne peut pas aboutir sans l'action du client
Sta tus Mov edP erm ane ntl y	en tie r	<u>301</u> - Redirection permanente.
Sta tus Fou nd	en tie r	<u>302</u> - Redirection temporaire.
Sta	en	<u>303</u> - Temporairement

tus See Oth er	tie r	déplacé.
Sta tus Not Mod ifi ed	en tie r	<u>304</u> - La ressource n'a pas été modifiée depuis la dernière requête possibilité d'utiliser la version en cache pour plus d'efficacité.
Sta tus Use Pro xy	en tie r	<u>305</u> - La ressource n'est accessible que par le biais du proxy spécifié dans l'en-tête.
Sta tus Tem por ary Red ire ct	en tie r	<u>307</u> - Redirection temporaire sans changement de méthode.
Sta tus Per man ent	en tie r	<u>308</u> - Redirection permanente sans changement de méthode.

Red ire ct		
Sta tus Bad Req ues t	en tie r	<u>400</u> - Syntaxe de requête erronée.
Sta tus Una uth ori zed	en tie r	<u>401</u> - Authentification incorrecte ou manquante.
Sta tus For bid den	en tie r	<u>403</u> - Accès refusé en raison d'un manque de permissions.
Sta tus Not Fou nd	en tie r	<u>404</u> - Ressource non trouvée.
Sta tus	en tie	<u>405</u> - Méthode non autorisée.

Method	Not Allowed	
Status	408 - Temps d'attente écoulé côté serveur.	
Status	409 - Conflit avec l'état actuel du serveur.	
Status	410 - Ressource supprimée, aucune redirection connue.	
Status	413 - Requête trop volumineuse.	

tit yTo oLa rge		
Sta tus Req ues tUR ITo oLo ng	en tie r	<u>414</u> - URI trop longue.
Sta tus Uns upp ort edM edi aTy pe	en tie r	<u>415</u> - Format non supporté.
Sta tus Unp roc ess abl eEn tit y	en tie r	<u>422</u> - Entité incompréhensible ou incomplète.

Sta tus Too Man yRe que sts	en tie r	<u>429</u> - Trop de requêtes dans un délai donné.
Sta tus Int ern als erv erE rro r	en tie r	<u>500</u> - Erreur interne serveur.
Sta tus Not Imp lem ent ed	en tie r	<u>501</u> - Fonctionnalité non supportée.
Sta tus Bad Gat ewa y	en tie r	<u>502</u> - Réponse invalide d'un serveur amont.

Sta tus Ser vic eUn ava ila ble	en tie r	<u>503</u> - Service indisponible.
Sta tus Gat ewa yTi meo ut	en tie r	<u>504</u> - Timeout d'un serveur amont.

Statuts HTTP occasionnels (valeurs
possibles de la propriété :Status)

Nom	Ty pe	Usage
Sta tus Con tin ue	en tie r	<u>100</u> - La requête est en cours de traitement.
Sta	en tie	<u>101</u> - Changement de protocole accepté.

tus r		
Swi tch ing Pro toc ols		
Sta tus Pro ces sin g	en tie r	<u>102</u> - Traitement en cours (évite un timeout côté client).
Sta tus Ear lyH int s	en tie r	<u>103</u> - Envoi anticipé de liens avant réponse définitive.
Sta tus Non Aut hor ita tiv eIn fo	en tie r	<u>203</u> - Contenu modifié par un proxy.

Sta tus Res etC ont ent	en tie r	<u>205</u> - Réinitialiser l'affichage côté client.
Sta tus Mul tis tat us	en tie r	<u>207</u> - Réponse multiple.
Sta tus Alr ead yRe por ted	en tie r	<u>208</u> - Déjà reporté dans cette collection.
Sta tus IMU sed	en tie r	<u>226</u> - Réponse encodée sous forme de delta.
Sta tus Mul tip leC	en tie r	<u>300</u> - URI correspondant à plusieurs ressources.

hoi ces		
Sta tus Use Pro xy	en tie r	<u>305</u> - Requête à ré- adresser via un proxy.
Sta tus Pay men tRe qui red	en tie r	<u>402</u> - Paiement requis.
Sta tus Not Acc ept abl e	en tie r	<u>406</u> - Aucun format acceptable selon les en-têtes Accept.
Sta tus Pro xyA uth Req uir	en tie r	<u>407</u> - Authentification proxy requise.

ed		
Status Length Required	entier	411 - Longueur de requête non précisée.
Status Precondition Failed	entier	412 - Précondition(s) non vérifiée(s).
Status Requested Range Not Satisfiable	entier	416 - En-tête Range invalide.

Sta tus Exp ect ati onF ail ed	en tie r	<u>417</u> - Expectation insatisfaite.
Sta tus Tea pot	en tie r	<u>418</u> - Je suis une théière (RFC 2324 / 1er avril).
Sta tus Pag eEx pir ed	en tie r	<u>419</u> - Ressource expirée.
Sta tus Bad Map pin g	en tie r	<u>421</u> - Requête envoyée à un serveur incapable de répondre (connexion réutilisée, etc.).
Sta tus Loc ked	en tie r	<u>423</u> - Ressource verrouillée.

Sta tus Met hod Fai lur e	en tie r	<u>424</u> - Échec dû à une autre opération (dépendance).
Sta tus Too Ear ly	en tie r	<u>425</u> - Trop tôt (risque de rejeu).
Sta tus Upd ate Req uir ed	en tie r	<u>426</u> - Mise à niveau requis (ex: upgrade vers TLS).
Sta tus Inv ali dDi git als ign atu re	en tie r	<u>427</u> - Signature numérique invalide.

Sta tus Pre con dit ion Req uir ed	en tie r	<u>428</u> - Requête doit être conditionnelle.
Sta tus Req ues tHe ade rFi eld sTo oLa rge	en tie r	<u>431</u> - En-têtes HTTP trop volumineux.
Sta tus NoR esp ons e	en tie r	<u>444</u> - Aucune réponse renvoyée, connexion fermée.
Sta tus Ret	en tie r	<u>449</u> - Réessayer après action préalable.

ryW ith		
Sta tus Blo cke dBy WPa ren tal Con tro ls	en tie r	<u>450</u> - Bloqué par contrôle parental Windows.
Sta tus Una vai lab leF orL ega lRe aso ns	en tie r	<u>451</u> - Indisponible pour raison légale.
Sta tus SSL Cer tif	en tie r	<u>495</u> - Certificat client invalide (extension).

ica teE rro r		
Sta tus SSL Cer tif ica teR equ ire d	en tie r	<u>496</u> - Certificat client requis (extension).
Sta tus HTT PRe que stS ent toH TTP SPo rt	en tie r	<u>497</u> - Requête HTTP envoyée vers port HTTPS (extension).
Sta tus Tok enE xpi	en tie r	<u>498</u> - Jeton expiré ou invalide.

red OrI nva lid		
Sta tus Cli ent Clo sed Req ues t	en tie r	<u>499</u> - Client a fermé la connexion avant réponse.
Sta tus HTT PVe rsi onN otS upp ort ed	en tie r	<u>505</u> - Version HTTP non supportée.
Sta tus Var ian tAl soN	en tie r	<u>506</u> - Erreur de négociation.

ego		
tia		
tes		
Sta	en	<u>507</u> - Espace
tus	tie	insuffisant.
Ins	r	
uff		
ici		
ent		
Sto		
rag		
e		
Sta	en	<u>508</u> - Boucle détectée.
tus	tie	
Loo	r	
pDe		
tec		
ted		
Sta	en	<u>509</u> - Dépassement de
tus	tie	quota/bande
Ban	r	passante.
dwi		
dth		
Lim		
itE		
xce		
ede		
d		

Sta tus Not Ext end ed	en tie r	510 - Requête ne respecte pas une politique étendue.
Sta tus Net wor kAu the nti cat ion Req uir ed	en tie r	511 - Authentification réseau requise (portail captif).

Méthodes

 **RecordToVariant**

 **Présentation**

Convertit l'enregistrement courant d'une source HFSQL en Variant.

 **Prototypage**

```
RecordToVariant(pSource Source
```

Paramètres

No m	T y p e	Usage
pS ou rc e	S o u rc e d e D o n n é e s	Source positionnée sur un enregistrement.
pC ol um ns	c h â n e	Liste de rubriques séparées par virgules, ou *.

Valeurs retournées

Ty pe	Usage
Variant	Objet Variant représentant l'enregistrement.

Exemple

- Lecture d'un enregistrement HFSQL (client recherché par ID)
- Conversion de l'enregistrement courant en Variant pour simplifier la sérialisation
- Renvoi du Variant au format JSON au client REST

```
FONCTION ClientGET(poRe  
objet lrRequest, poResponse  
objet lrResponse)
```

```
vClient est Variant  
sID      est chaîne =  
poRequest:GetRouteValue("id"  
)
```

```
SI sID="" ALORS  
    poResponse:Status      =  
    lrResponse::StatusBadRequest  
    poResponse:ContentType =  
    lrResponse::ContentTXT  
    poResponse:Body        =  
    "No ID provided"
```

[Copier](#)

```
RETOUR
FIN

SI
hLitRecherchePremier("CLIENT
", "ID", sID) ALORS
    vClient =
    lrResponse::RecordToVariant(
    CLIENT,
    "ID,NAME,FIRSTNAME,CITY")

    poResponse:Status      =
    lrResponse::StatusOK
    poResponse:ContentType =
    lrResponse::ContentJSON
    poResponse:Body        =
    VariantVersJSON(vClient)
SINON
    poResponse:Status      =
    lrResponse::StatusNotFound
    poResponse:ContentType =
    lrResponse::ContentTXT
    poResponse:Body        =
    "Client not found"
FIN
```



SetBinaryContent



Présentation

Configure une réponse binaire (:Body=buffer, :IsBinary=Vrai), et renseigne le membre :ContentType si fourni



Prototypage

```
SetBinaryContent(pContent Buffer
```

Paramètres

No m	T y p e	Usage
pC on te nt	B u ff er	Contenu binaire
pC on te nt Ty pe	c h aî n e	MIME type (optionnel).

Valeurs retournées

(aucune)

Exemple

- Renvoi d'un fichier binaire (ex: image PNG) depuis un endpoint REST

- Lecture du fichier dans un Buffer
- Remplissage automatique de :Body et :IsBinary via SetBinaryContent

```
FONCTION LogoGET(poRequ
objet lrRequest, poResponse
objet lrResponse)

bufImg est Buffer

bufImg =
fChargeBuffer("c:\img\logo.p
ng")

SI Taille(bufImg)=0 ALORS
    poResponse:Status =
lrResponse::StatusNotFound
    poResponse:ContentType =
lrResponse::ContentTXT
    poResponse:Body =
"File not found"
    RETOUR
FIN

poResponse:SetBinaryContent(
bufImg,
lrResponse::ContentPNG)
poResponse:Status =
lrResponse::StatusOK
```

[Copier](#)

SetHeaderValue



Présentation

Affecte une valeur à un header de réponse.

 Prototypage

```
SetHeaderValue(pTag chaîne, pVa
```

 Paramètres

No m	T y p e	Usage
pT ag	c h â î n e	Nom du header.
pV al ue	c h â î n e	Valeur.

 Valeurs retournées

(aucune)

 Exemple

- Ajout d'un header HTTP applicatif (ex:

identifiant de corrélation pour tracer une requête)

- Renvoi d'une réponse simple "OK"
- Pratique pour diagnostiquer des appels côté client (log, monitoring, support)

```
FUNCTION Ping(poRequest  
objet lrRequest <utile>,  
poResponse objet lrResponse)
```

[Copier](#)

```
sReqID est chaîne =  
DonneUUID()
```

```
poResponse:SetHeaderValue("X-Request-ID", sReqID)
```

```
poResponse:Status      =  
poResponse::StatusOK  
poResponse:ContentType =  
poResponse::ContentTXT  
poResponse:Body        =  
"pong"
```

```
RENOYER poResponse
```

SourceToVariant

Présentation

Convertit une source HFSQL en tableau de Variants.

Prototypage

```
SourceToVariant (pSource Source
```

Paramètres

No m	T y p e	Usage
pS ou rc e	S o u rc e d e D o n n é e s	Source HFSQL.
pC ol um ns	c h aî n e	Liste de rubriques séparées par virgules, ou *.

Valeurs retournées

Ty pe	Usage
----------	-------

Var ian t	Liste (tableau) de Variants.
-----------------	------------------------------

Exemple

- Liste des enregistrements d'une table HFSQL (ex: CLIENT)
- Conversion de l'ensemble en tableau de Variants
- Renvoi au format JSON (endpoint REST de type /clients)

```
FONCTION
ClientsGET(poRequest objet
lrRequest <utile>,
poResponse objet lrResponse)

vClients est Variant

SI hNbEnr(CLIENT)=0 ALORS
    poResponse:Status =
lrResponse::StatusNotFound
SINON
    vClient =
    lrResponse::SourceToVariant(
    CLIENT,
    "ID,NAME,FIRSTNAME,CITY")

    poResponse:Status =
lrResponse::StatusOK
    poResponse:ContentType =
lrResponse::ContentJSON
```

[Copier](#)

```
poResponse:Body =  
VariantVersJSON(vClients)  
fin
```

© CODE LINE 2018-2026

