

Stratus Agent

Stratus Agent

Stratus Agent is a web based solution that assists insurance companies and surplus lines brokers to more accurately pinpoint Kentucky policy locations in real time. Agent is a point of sale service that integrates with a company's existing policy management systems to meet statutory requirements defined in KRS 91A.080 – 91A.812 for the collection and reporting of local premium taxes for 367 municipalities and 44 counties in Kentucky. Agent uses geocoding technology to return a matching address and plots the address to a digital map of Kentucky. Agent is a Kentucky verified risk location system.

Stratus Agent offers all the key features of the previous platform, but has been redesigned using the latest technology with both security and efficiency in mind. The new web service is more robust and is housed in a state of the art cloud infrastructure. The code is developed using security based standards. All these lend itself to improved performance, seamless updates, and a better overall experience.

What We'll Cover

- Overview
- GCP (Cloud Infrastructure)
- Auth0 (Access Management and Authentication)
- Precisely (Geocoding Engine)
- SOAP (WSDL)
- REST
- XML Results
- User Interface – COMING SOON!

Overview

Stratus Agent is a point-of-sale service that can be quickly and easily integrated into your company's existing website or policy/agent management system, or you can log-in and begin using the TriTech web-interface (Web interface COMING SOON).

Using SOAP or REST (NEW!), you can pass a valid address and premium type code (line of business) to Agent, the service will return a matching address, tax jurisdiction, and tax rate information.

Cloud Infrastructure

Stratus agent leverages the reliable, secure and highly scalable cloud computing services provided by Google (GCP). GCP <https://cloud.google.com/security>

Access Management and Authentication

Stratus Agent uses Auth0 to store, manage and authenticate users. Using the latest technology, we are able to offer federated logins, SSO, and MFA. <https://auth0.com/docs/>

Geocoding

Stratus Agent uses a third party geocoding engine, Precisely, to accurately determine the location of an address. Addresses are converted to coordinates and then plotted to map with the appropriate municipal boundaries.

SOAP API

Before you can utilize the Stratus Agent Web Service/API, you must obtain a company ID and password from your TriTech representative for the SOAP API call.

For a formal definition of the web service, please see the provided WSDL at

https://tritechsoft.com/public_download/StratusAgent.wsdl

Please note, if you are a current Agent Client using SOAP, you do NOT need to make any changes to your system.

REST API (M2M)

In lieu of a company ID and password, you will be provided with a Client ID and Client Secret to invoke the Agent REST API. An access token must be obtained before your application can gain access to TriTech's Stratus Agent M2M resources.

There are two endpoints for the REST API.

<http://agent.tritechsoft.com/api/token>

<http://agent.tritechsoft.com/api/allocate>

The /api/token endpoint returns a JSON Web Token (JWT) which is required for the /api/allocate endpoint. It expects a clientId and clientSecret and returns a token string; see example below.

URL: <http://agent.tritechsoft.com/api/token>

HTTP Method: POST

Request Body:

```
{
  "clientId":"string",
  "clientSecret":"string"
}
```

Response: "ACCESS_TOKEN"

Response status codes:{

```
"200":{
  "description":"OK",
},
"401":{
  "description":"Unauthorized"
},
"500":{
```

```
        "description": "Internal server error"
    }
}
```

You must have an active, valid account registered with TriTech to acquire a JWT.

The token must be used for the call to the /api/allocate endpoint.

The token has a valid lifetime of 86400 seconds; call /api/token on expiration to acquire a new one.

URL: <http://agent.tritechsoft.com/api/allocate>

HTTP Method: POST

HTTP Header:

```
    "key": "Authorization",
    "value": "Bearer ACCESS_TOKEN",
```

Request Body:

```
{
    "effectiveDate": "string",
    "lineOfBusiness": "string",
    "sourceCity": "string",
    "sourceState": "string",
    "sourceStreet": "string",
    "sourceZipCode": "string"
}
```

Response body:

```
{
  "allocateAddressResult": {
    "censusBlockID": "string",
    "censusCountyCode": "string",
    "censusCountyName": "string",
    "cityCode": "string",
    "cityName": "string",
    "cityOrCounty": "string",
    "cityTaxRates": {
      "casualty": 0,
      "fire": 0,
      "flatFee": 0,
      "health": 0,
      "inlandMarine": 0,
      "life": 0,
      "minTax": 0,
      "motorVehicle": 0,
      "other": 0
    },
    "countyCode": "string",
    "countyCreditApplies": true,
    "countyFipsCode": "string",
```

```

"countyName": "string",
"countyTaxRates": {
  "casualty": 0,
  "fire": 0,
  "flatFee": 0,
  "health": 0,
  "inlandMarine": 0,
  "life": 0,
  "minTax": 0,
  "motorVehicle": 0,
  "other": 0
},
"effectiveCityRate": 0,
"effectiveCountyRate": 0,
"effectiveDate": "string",
"effectiveTaxRate": 0,
"errorDetails": {
  "description": "string",
  "number": 0,
  "source": "string"
},
"lineOfBusiness": "string",
"matchQuality": {
  "matchCode": 0,
  "matchDescription": "string",
  "matchIdentifier": "string"
},
"matchedAddress": {
  "city": "string",
  "state": "string",
  "street": "string",
  "zipCode": "string"
},
"sourceAddress": {
  "city": "string",
  "state": "string",
  "street": "string",
  "zipCode": "string"
},
"taxCode": "string"
}
}

```

```

Response status codes:{
  "200":{
    "description":"OK",
  },
  "401":{
    "description":"Unauthorized"
  }
}

```

```
    },
    "403":{
        "description":"Forbidden"
    },
    "500":{
        "description":"Internal Server Error"
    }
}
```

XML Elements Returned

Valid element names are bolded:

1. **SourceAddress** - The address you typed in.
 - **Street**
 - **City**
 - **State**
 - **ZipCode**
2. **LineOfBusiness** – The original line of business code you entered will be returned.
3. **MatchedAddress** –An address is returned from the address database, if a close match to your source address can be found.
 - **Street**
 - **City**
 - **State**
 - **ZipCode**

If an address was not returned, it is due to the following reasons:

- The address is a PO Box.
 - The address is a Rural Route or Highway Contract Route.
 - A similar address could not be found. Either the address is invalid or a similar address could not be found in the database.
4. **CensusBlockID** – The Census Block ID of the **MatchedAddress** contains relevant US Census information.
 5. **CountyFIPSCode** – The county FIPS code is derived from the **CensusBlockID**. The first 2 digits identify the state jurisdiction and the last 3 digits identify the county jurisdiction of the **MatchedAddress**.
 6. **CensusCountyName** – The county of the **MatchedAddress** is determined by plotting address coordinates to a Kentucky map. **CensusCountyName** is always returned; *it does not* indicate that the county taxes the policy.

7. **CensusCountyCode** – If **CensusCountyName** has a corresponding tax code in the KY Annual Insurance Bulletin, it will be returned. This *does not indicate* the county taxes the policy. If there is no tax code, the number 9998 will be returned.

8. **MatchQuality** – In order for a **MatchedAddress** to be returned, **MatchQuality** must be rated as a close match. Precisely’s geocoding engine uses its own algorithm to determine what a close match is. The street or house number of a **MatchedAddress** must be identical to the number in the **SourceAddress** to be considered a close match. Other elements of the address are evaluated to determine if the **MatchedAddress** is a good candidate for the **SourceAddress** provided.
 - Good Match. 1 close match was returned.
 - No Close Match. An address closely corresponding to the source address was not found. No **MatchedAddress** will be returned.
 - PO Box or Rural Route. The **SourceAddress** or **MatchedAddress** was determined to be a PO Box, Rural Route, or HC Route. No **MatchedAddress** will be returned.

9. **MatchCode** –
 - 0 indicates a good match.
 - 2 indicates no close match.
 - 4 indicates a PO Box, RR, or HC Route.

10. **MatchIdentifier** –Indicates which parts of a **MatchedAddress** are an exact match to its **SourceAddress**. The **MatchIdentifier** contains 10 characters that represent different parts of the address. If Agent is able to match part of an address, a character will appear in the match identifier at the position corresponding to that part of the address. If not, a dash will appear instead. The **MatchIdentifier** positions are as follows:

Geocode Type

The first character in the **MatchIdentifier** is always: S – Street Type

Positional Accuracy

The second character indicates the positional accuracy of the candidate’s point.

Number	Positional Accuracy
8	A single point represents the address
7	Interpolated points along a street segment/parcel
6	Centroid of a site or building with a dedicated zip code
5	Interpolated points along a street segment
4	Street centroid
3	ZIP + 4 centroid
2	ZIP + 2 centroid

1	ZIP centroid
0	No coordinates available

Address Position

These positions describe the matches in the different elements in the address.

Position	Letter Code	Address Position
1	H	House Number
2	P	Street Prefix
3	N	Street Name
4	T	Street Type
5	S	Street Suffix
6	C	City Name
7	Z	Zip Code

Example: The **MatchIdentifier** S5HPN-S-Z is interpreted as follows:

- The point is a street type known as an interpolated street segment.
- The house number, street prefix, street name, street suffix and zip code of the **MatchedAddress** are identical to the **SourceAddress**.
- The street type and city name in the **MatchedAddress** are not identical to the **SourceAddress**. This could be due to spelling or it may be completely different.

11. **CityCode** –This corresponds to the City Code that appears on the KY Dept of Insurance Local Government Premium Tax Schedule. When a **MatchedAddress** is returned, Agent plots coordinates to a map to determine the taxing jurisdiction.
 - If the address falls within the boundaries of a city that taxes policies, the **CityCode** for the city is returned.
 - If the address falls outside a city that taxes or inside a city that does not tax, the address is assigned to the county. The county **CityCode** is returned.
 - If a **CityCode** is not returned in the above 2 steps; a code of 9998 will be returned. This indicates that the policy is not taxable.
12. **CityName** –The name of the city or county corresponding to the **CityCode** will be returned. **CityName** may or may not match the city found in the **MatchedAddress**. This is the actual physical location, not the name listed in the physical address. The word “NONTAXABLE” will be returned if a **CityCode** of 9998 is returned.

13. **CityOrCounty** – If the **CityCode** and **CityName** returned are for a city, the word *City* will be returned. If the code and name are for a county, the word *County* will be returned. If the policy is nontaxable, nothing will be returned.
14. **TaxCode** –**TaxCode** is returned for any municipality or county that has one. It is listed on Kentucky Local Government Premium Tax Schedule and the Local Government Premium Tax Code Descriptions. The maximum length for this element is 10.
15. **CityTaxRates** – The tax rates for each line of business will be returned for the **CityCode**. If the **CityCode** is 9998, a tax rate of 0 will be returned. These rates are from the Kentucky Local Government Premium Tax Schedule.
- **Casualty**
 - **Fire**
 - **Health**
 - **Inland Marine**
 - **Life**
 - **Motor Vehicle**
 - **Other**
 - **MinTax** (Minimum Tax will be returned if applicable. A minimum tax cannot be assessed on a policy by policy basis. It can only be assessed on total premiums)
 - **FlatFee** (A flat fee cannot be assessed on a policy by policy basis. It can only be assessed on total premiums)
16. **CountyCreditApplies** – If premiums taxable by a city are also taxable by a county, *true* will be returned. Otherwise *false* will be returned. This applies to cities with a **TaxCode** of *A* or *B*.
17. **CountyCode** – If **CountyCreditApplies** is *true*, the county's **CityCode** will be returned.
18. **CountyName** – If **CountyCreditApplies** is *true*, the county's **CityName** will be returned.
19. **CountyTaxRates** – If **CountyCreditApplies** is *true*, the county's **CityTaxRates** will be returned.
- **Casualty**
 - **Fire**
 - **Health**
 - **Inland Marine**
 - **Life**

- **Motor Vehicle**
 - **Other**
 - **MinTax** (Minimum Tax will be returned if applicable. A minimum tax cannot be assessed on a policy by policy basis. It can only be assessed on total premium),
 - **FlatFee** (A flat fee cannot be assessed on a policy by policy basis. It can only be assessed on total premium).
20. **EffectiveDate** – Returns the **EffectiveDate** you entered. Otherwise the date is the current date. Tax rates returned are from this period.
21. **EffectiveTaxRate** – An **EffectiveTaxRate** is returned for the **LineOfBusiness** you specified. If you do not specify a **LineOfBusiness**, the **EffectiveTaxRate** is zero. The **EffectiveTaxRate** rate may be multiplied by the premium amount to determine the total tax amount to be passed to the insured.
- If **CountyCreditApplies** is *false*, the **EffectiveTaxRate** is pulled from **CityTaxRates**,
 - If **CountyCreditApplies** is *true*, the **EffectiveTaxRate** is pulled from **CityTaxRates** or **CountyTaxRates** (whichever is higher).
 - If **CountyCreditApplies** is *true*, but the city name is Springfield, Hickman, or Lawrenceburg, the effective tax rate will be the sum of the **CityTaxRates** and **CountyTaxRates**.
22. **EffectiveCityRate** – The **EffectiveCityRate** is returned for the **LineOfBusiness** you specified. This is pulled from **CityTaxRates**. The **EffectiveCityRate** can be applied to premiums to determine the tax payable to the city or county.
- If **CountyCreditApplies** is *false*, the **EffectiveCityRate** is the same as the **EffectiveTaxRate**. may be multiplied by the premiums to determine
 - If **CountyCreditApplies** is *true*, the **EffectiveCityRate** is the same as the **CityTaxRates**.
23. **EffectiveCountyRate** – This is the **EffectiveTaxRate** less the **EffectiveCityRate**. The **EffectiveCountyRate** can be applied to premiums to determine the tax due to the county ONLY when both the county and city tax the same premiums.
24. **ErrorDetails** –
- If no error occurs, **Description** remains blank and **Number** returned is 0.
 - Error **Description** and **Number** for errors are:
 - Geocoder Server error, 1000
 - No CompanyID was specified, 1601
 - No Password was specified, 1602
 - Invalid CompanyID, 1603
 - Incorrect Password, 1604
 - Account suspended, 1605
 - Tax rate not available for this date, 1606

- Incorrect Premium Type Code, 16071
- Premium Type Code Missing, 16072
- Unspecified Error, 1600

User Interface

Coming Soon!